



Federal Aviation Administration

U.S. Department of Transportation
Federal Aviation Administration
NAS Voice System

FAA NVS Addendum
to
EUROCAE ED-137A Part III

**Voice Recording/Event Logging
Requirements
and
Specification**

Federal Aviation Administration
800 Independence Avenue SW
Washington, DC 20591

TABLE OF CONTENTS

1.0 INTRODUCTION.....	8
1.1 Scope of document	8
2.0 DEFINITIONS	9
2.1 Terminology for requirements, recommendations and options	10
2.2 Version Tracking	10
3.0 RECORDING REQUIREMENTS	11
3.1 Legal Recording Requirements	11
3.2 Voice recording entities	11
3.3 Voice location identification.....	11
3.4 Active Recording	11
3.5 Recording of all Ground-Ground audio communications at position level.....	12
3.6 Recording of Ground-Ground audio communications at the VCS endpoint.....	12
3.7 Recording of all Air-Ground audio communications at position level.....	13
3.8 Recording of Air-Ground audio communications at the VCS endpoint.....	13
3.9 Recording of Air-Ground audio communications at the Radio/Radio Gateway endpoint	14
3.10 Provision for 2 identical autonomous voice recordings at VCS endpoint.....	15
3.11 Provision for dual RTSP sessions for position recording.....	15
3.12 Provision for at least one voice recording at Radio or Radio Gateway endpoint.....	15
3.13 True and faithful representation of audio signal.....	15
3.14 Date & Timestamp Synchronization for audio	16
3.15 Date & Timestamp Synchronization for Call Record Data	17
3.16 Voice Packet size towards RTSP server.....	17
3.17 Diffserv codepoint for Recorder audio	17
3.18 Audio codecs	18
3.18.1 Recording compressed audio streams	18
3.18.2 Recording uncompressed audio streams	18
3.18.3 Audio codec request to RTSP server.....	19
3.18.4 Audio encryption over RTSP session.....	19
3.19 SIP.....	19
3.19.1 SIP session establishment with Recorder.....	19
3.19.2 Registration with SIP Registrar Server	19
3.19.3 Transport protocol.....	19
3.20 Transport protocol	20

3.21	RTP	20
3.21.1	Transport of audio	20
3.21.2	Transport of RTP header extension	21
3.22	RTP with RTCP	21
3.23	RTSP	21
3.23.1	Transport of messages	21
3.23.2	RTSP server port	21
3.23.3	RTSP addresses	21
3.23.4	RTSP address permissions list	22
3.23.5	OPTIONS request message	22
3.23.6	RTSP session establishment for G/G communications	22
3.23.7	RTSP session termination for G/G communications	23
3.23.8	RTSP session establishment for A/G communications	23
3.23.9	RTSP session terminated for A/G communications	25
3.23.10	Lost audio prevention during RTSP session	25
3.23.11	Lost audio prevention during RTSP session teardown	25
3.24	Bandwidth to RTSP server	26
3.24.1	RTSP session timeout	26
3.25	RTSP server session limit	26
3.26	Client session limit	26
3.27	Referencing G/G communications	27
3.27.1.1	RTSP Session Identity Reference value	27
3.27.1.2	Connection Reference value	27
3.27.1.3	Call Reference value	28
3.27.1.4	Thread Reference value	28
3.28	Referencing A/G communications	29
3.28.1.1	RTSP Session Identity Reference value	29
3.28.1.2	Connection Reference value	29
3.28.1.3	Thread Reference value	30
3.29	Call Record Data for G/G communications	31
3.29.1	Call Record Data Properties for G/G communications	31
3.29.2	Call Record Data Operations for G/G communications	32
3.29.3	Call intrusion and conference call management towards RTSP server	32
3.29.4	Call Hold management towards RTSP server	32
3.30	Call Record Data for A/G communications	33
3.30.1	Call Record Data Properties for A/G communications	35
3.30.1.1	Call Record Data Operations for A/G communications	35
3.30.1.2	PTT operation for A/G communications	36
3.30.1.3	SQU operation for A/G communications	36
3.30.2	Management of Simultaneous aircraft calls on different frequencies towards RTSP server	36
3.30.3	Management of Receiver Voting operation towards RTSP server	37
3.30.3.1	Receiver voting for A/G communications	37
3.31	Replay	37

3.32	RTSP server generation of CRD statistics	38
3.33	RTSP server archiving	38
4.0	RECORDING SPECIFICATION	39
4.1	Scope of specification	39
4.2	Applicable configurations	39
4.3	Basic Protocol requirements	39
4.4	Audio Specifications	40
4.5	Active Recording	40
4.6	Dual RTSP sessions from position	40
4.7	Date & Timestamp Synchronization for audio	41
4.8	RTSP addresses	41
4.8.1	RTSP address permissions list	41
4.9	OPTIONS message	42
4.10	Recording Model for Telephone communication	42
4.11	Recording Model for Radio communication	43
4.11.1	RTP Header Extension	44
4.12	Session Initiation Protocol (SIP)	44
4.12.1	Logical SIP Entities	45
4.13	RTSP	45
4.14	Transport	46
4.14.1	RTP over independent TCP	47
4.14.2	Framing Method	48
4.14.3	Lost audio prevention during RTSP session teardown	49
4.14.4	RTSP session timeout	49
4.15	RTSP Control Messages	49
4.15.1	ANNOUNCE and SETUP	49
4.15.2	RECORD	51
4.15.3	PAUSE	52
4.15.4	SET_PARAMETER	52
4.15.5	TEARDOWN	53
4.15.6	REPLAY	54
4.15.7	RTSP session signaling procedures between clients and RTSP server	56
4.16	Call Record Data Format	58
4.16.1	Properties	58
4.16.2	Operations	59
4.16.3	Time Stamp format	60
4.17	Referencing Call Scenarios	60
4.17.1	Connection Reference generation	60
4.17.2	Call Reference generation (Telephone only)	60
4.17.3	Thread Reference generation	61

4.17.4	ConfRef generation for conference	61
4.17.5	CRD XML body in RECORD, PAUSE, SET_PARAMETER and TEARDOWN	61
4.17.6	SETUP without Call Record Data.....	62
4.18	WG67-Version.....	62
4.19	Telephone Call Record Data.....	63
4.19.1	Telephone Call Record Data Properties	63
4.19.2	Telephone Call Record Data Operations.....	64
4.19.3	CRD signaling procedures between clients and RTSP server for G/G communications	65
4.19.4	CRD signaling procedures between clients and RTSP server for a conference	66
4.20	Radio Call Record Data	67
4.20.1	Radio Call Record Data Properties	68
4.20.2	Radio Call Record Data Operations	68
4.20.3	CRD signaling procedures between clients and RTSP server for A/G communications	71
ANNEX A : CALL RECORD DATA MESSAGE EXAMPLES.....		72
A.1.	CRD messages for Basic Telephone Calls	72
A.1.1.	Basic outgoing call – unanswered example	72
A.1.2.	Basic incoming call – answered example.....	72
A.1.3.	Basic outgoing call – answered example	73
A.1.4.	Recording a rejected call example.....	74
A.1.5.	Unattended Call Transfer example	74
A.1.6.	RTSP message sequence for Call intrusion description.....	76
A.1.7.	Call intrusion example	77
A.1.8.	Conference call example	78
A.1.9.	Call Hold operation example.....	79
A.2.	CRD messages for Radio Calls.....	81
A.2.1.	PTT activation at VCS endpoint example	81
A.2.2.	PTT activation at Radio endpoint example	81
A.2.3.	SQU activation at VCS endpoint example	82
A.2.4.	SQU activation at Radio endpoint example	82
A.2.5.	Simultaneous SQU received at VCS endpoint on different frequencies	83
A.2.6.	Receiver Voting operation	84
ANNEX B : REFERENCES		86
ANNEX C : ACRONYMS.....		88

LIST OF FIGURES

FIGURE 1: RECORDING TELEPHONE COMMUNICATION	42
FIGURE 2: RECORDING RADIO COMMUNICATION	43
FIGURE 3: SIP SESSION ESTABLISHMENT: MESSAGE SEQUENCE	45
FIGURE 4: RTSP RECORD AND REPLAY SESSION	46
FIGURE 5: TCP FRAME FORMAT	48
FIGURE 6: RTSP SESSION ESTABLISHMENT: MESSAGES AND SEQUENCE	50
FIGURE 7: RECORD: MESSAGES AND SEQUENCE.....	51
FIGURE 8: PAUSE: MESSAGES AND SEQUENCE	52
FIGURE 9: SET_PARAMETER: MESSAGES AND SEQUENCE	53
FIGURE 10: TEARDOWN MESSAGES AND SEQUENCE.....	53
FIGURE 11: RTSP REPLAY SESSION: MESSAGES AND SEQUENCE	55

LIST OF TABLES

TABLE 1- LIST OF TELEPHONE PROPERTIES	63
TABLE 2- CALL TIMER PROPERTIES AND MEANING	64
TABLE 3- LIST OF TELEPHONE OPERATIONS	64
TABLE 4- LIST OF RADIO PROPERTIES	68
TABLE 5- LIST OF RADIO OPERATIONS (1)	68
TABLE 6- LIST OF RADIO OPERATIONS (2)	69
TABLE 7- LIST OF RADIO OPERATIONS (3)	69
TABLE 8- RRC CONTROL BITS	70

1.0 INTRODUCTION

1.1 Scope of document

The scope of this document is to define the FAA addendum for Recording requirements and specification.

It has been based on the current definition of the EUROCAE ED-137A Recording document, the FAA VoIP-IE Guidelines for Recording tests, the configuration employed in order to achieve interoperability of Recorder implementations with VCS and Radio equipment demonstrated during the FAA VoIP-IE and finally the Recording recommendations defined in the FAA VoIP-IE final report.

In case of any resulting conflict in specification between the EUROCAE ED-137A Recording document and this FAA addendum for Recording requirements and specification, the contents of this document **SHALL** prevail.

Following the FAA VoIP-IE, detailed feedback was provided to EUROCAE WG67 with the approval of all FAA VoIP-IE participating vendors regarding the implementation of the Recorder specification for all issues identified during the event that would improve clarity, correct or generally improve the specification. Some of these issues have been agreed to by the WG67 group for integration within the next edition of the ED-137 Volume 3 Recording specification.

Although the specification is defined for a single recorder, the protocol and functionality can be applied to multiple recorders.

Chapter 3: Defines the Recording requirements

Chapter 4: Defines the Recording specification

Annex A: Defines Call Record Data message examples

The specification is applied to legal recorders connected to the User Terminals (Mandatory), recorders connected to VCS network interfaces for G/G communications and A/G channels (Recommended) and recorders connected to the radios for A/G communications (optional).

The specification follows the ED-137 Volume 3 Recording document, but has been enhanced to include Call Record Data relating to current editions of the ED-137A Volume 1 Radio and ED-137A Volume 2 Telephone. It therefore includes Call Record Data relating to the Radio Remote Control equipment and other RTP header extension control bits as well as the Override Call type.

2.0 DEFINITIONS

Active Recording: any client that sends or receives media streams (i.e. audio) during an established communication is responsible for sending a copy of these audio streams to the recorder(s).

Call Record Data: This is a data record produced by a VCS and Radio clients documenting the details of a communication that passed through the equipment. A Call Record Data is composed of properties and operations.

Legal Recorder - a system used in ATC facilities to record, store, archive, retrieve, playback, delete, and manage records of A/G and G/G voice communications between air traffic controllers and pilots and between air traffic controllers located at different facilities.

Operations: are events during the lifetime of a connection that may happen at any time and should be preserved at the recorder

Party (A-party, B-party, C-party and D-party): The users involved, sequentially, in a telephone call as follows:

A-party: the user who initiates a call (the calling party);

B-party: the user who first receives the call (the called party);

C-party: the third user involved in the call;

D-party: the fourth user involved in the call;

Properties: are single values that will not change during the lifetime of a connection and usually do not require a time reference, except for properties that are representing timestamp information.

2.1 Terminology for requirements, recommendations and options

The terminology for requirements, recommendations and options in this document is based on RFC 2119 [4], which specifies Best Current Practice regarding the use of Key Words for the Internet Community. As such, the following terminology is applied:

- The word **SHALL** denotes a mandatory requirement;
- The word **SHOULD** denotes a recommendation;
- The word **MAY** denotes an option.

To avoid confusion with their natural meanings in the English language, the words **SHALL**, **SHOULD**, and **MAY** take on the meaning stated above only where printed in boldface. When printed in normal (Arial) typeface, the natural English meaning is meant.

Detailed description of terminology:

- SHALL:** This word has the same meaning as the phrase "REQUIRED" and means that the definition is an absolute requirement of the specification.
- SHALL NOT:** This phrase means that the definition is an absolute prohibition of the specification.
- SHOULD:** This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT:** This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY:** This word, or the adjective "OPTIONAL", mean that an item is truly optional.

2.2 Version Tracking

The Recorder implementation from the EUROCAE ED-137B Volume 4 document onwards, **SHALL** be referenced as recorder.01 in the WG67-Version RTSP header field as described in par. 4.18.

3.0 RECORDING REQUIREMENTS

3.1 Legal Recording Requirements

The recording of ATC communications **SHALL** adhere to ICAO –Annex 10, Volume II, Chapter 3.5 [30]

3.2 Voice recording entities

The Voice Recording system **SHALL** provide a means to record voice messages at a specific controller working position (Legal recording).

The Voice Recording system **SHOULD** provide a means to record voice messages at the following entities:

- a specific G/G interface (Transmit and Receive paths) at a VCS endpoint;
- a specific A/G interface (Tx path) at a VCS endpoint;
- a specific A/G interface (Rx path) at a VCS endpoint.

The Voice Recording system **MAY** provide a means to record voice messages at the following entities:

- a specific radio receiver or radio gateway (optional);
- a specific radio transceiver or radio gateway (optional);
- a specific 3rd party.

3.3 Voice location identification

The precise location of the recording point at the above entities **SHALL** be clearly identified and noticeable with a high integrity during replay of the audio recordings.

3.4 Active Recording

Recording **SHALL** be based on an active session opened from either a User Terminal client, VCS network interface client, Radio Receiver/transceiver/Gateway network interface client or specific 3rd party device to the RTSP Server (Recorder equipment).

3.5 Recording of all Ground-Ground audio communications at position level

In the case of client being located at the User Terminal, each User Terminal client with an established G/G communication (i.e. DA, IDA, IA or OVR) or concurrent G/G communications (i.e. DA and incoming OVR) **SHALL** generate a single RTP audio stream containing the sum of the incoming (IN) and outgoing (OUT) audio streams at the position relevant to a G/G communication. It **SHALL** establish dual RTSP sessions with its nominated RTSP server per G/G communication for redundancy purposes. The summed RTP audio streams being transported over each RTSP session **SHALL** be identical.

Note 1.

A DA/IDA/IA call between User Terminal clients for example will result in both clients performing the same action of summing its incoming and outgoing audio streams and then sending its audio stream towards its nominated RTSP server.

Note 2.

The incoming audio stream to a User Terminal client in the case of an established outgoing IA call will include summed audio from any active G/G and A/G communications at the called User Terminal client in the case that monitoring is enabled, while the incoming audio stream to a User Terminal client in the case of an established incoming IA call will include the calling user microphone only.

Note 3.

Simultaneous IA calls from 'A'-party to 'B'-party and from 'B'-party to 'A'-party are treated as two independent IA calls and hence will result in each User Terminal client sending two separate summed audio streams to its nominated RTSP server.

Note 4.

The incoming audio stream to a User Terminal client in the case of an established outgoing OVR call will always include summed audio from any active G/G and A/G communications at the called User Terminal client plus the called user microphone. While the incoming audio stream to a User Terminal client in the case of an established incoming OVR call will always include summed audio from any active G/G communications at the calling User Terminal client plus the calling user microphone. It is necessary for each User terminal client involved with an OVR call to sum its incoming stream(s) and outgoing stream and send one summed audio stream to its Recording Server.

3.6 Recording of Ground-Ground audio communications at the VCS endpoint

In the case of client being located at the VCS endpoint, VCS network interface clients with an established G/G communication (i.e. DA, IDA, IA or OVR) **SHOULD** generate a single RTP audio stream containing the sum all incoming (IN) and outgoing (OUT) audio streams at the interface relevant to the G/G communication. It **SHOULD** establish a single RTSP session with its nominated RTSP server per G/G communication and **MAY** establish dual RTSP sessions with its nominated RTSP server for redundancy purposes.

Note 5.

Recording at a VCS telephone network interface client does confirm that controller speech was sent to the network, but isn't confirmation that the Controller at the position actually listened to the audio recorded at this recording point. Recording at this point relates to audio of an individual call and can be useful to demonstrate in case of an audio path failure within the VCS for example.

3.7 Recording of all Air-Ground audio communications at position level

In the case of client being located at the User Terminal, each User terminal client with an established A/G communication **SHALL** generate a single RTP audio stream containing the sum of all of the outgoing (Tx) audio stream and the incoming (Rx) audio stream for all frequencies configured at the position. It **SHALL** establish dual RTSP sessions with its nominated RTSP server for redundancy purposes. The summed RTP audio stream **SHALL** be transported over both RTSP sessions towards its nominated RTSP server. The RTP audio streams being transported over each RTSP session **SHALL** be identical.

If local side tone (generated by the VCS) or remote side tone (Rx audio that contains off-air transmitted audio) is enabled within the configuration, the original Tx audio **SHALL** be summed twice.

If side tone (local or off-air) is not enabled then it **SHALL NOT** be summed.

In the case that Receiver Voting is configured at the position with multiple incoming (Rx) audio streams for the same frequency, the Voting algorithm would select just one audio stream.

In the case of Transmit only configurations (i.e. ATIS), the outgoing (Tx) audio stream **MAY NOT** be summed with the incoming (Rx) audio stream, if there is no associated Radio Receiver.

Note 6.

Transmitted audio sent back on the receive path can also be used to generate a side tone to the controller earpiece. Systems can be configured for remote side tone received off-air from the receiver, generation of a local side tone or no side tone generation.

3.8 Recording of Air-Ground audio communications at the VCS endpoint

In the case of a client being located at the VCS endpoint, VCS network interface clients with an established A/G communication **SHOULD** generate a single RTP audio stream that **MAY** contain one of the following:

- The outgoing (Tx) audio stream only, towards a Radio transmitter or Radio Gateway for a specific frequency;
- The incoming (Rx) audio stream only, from a Radio receiver for a specific frequency;
- The sum of the outgoing (Tx) audio stream and the incoming (Rx) audio stream, in the case of a Radio transceiver for a specific frequency;

The VCS network interface **SHOULD** establish a single RTSP session with its nominated RTSP server and **MAY** establish dual RTSP sessions with its nominated RTSP server for redundancy purposes.

Note 7.

Recording at a VCS radio network interface client does confirm that controller speech was sent to the network, but isn't confirmation that the Controller at the position actually listened to the audio recorded at this recording point. Recording at this point can be useful to demonstrate in case of an audio path failure within the VCS for example.

Note 8.

Some ANSPs have a requirement to record audio on the Tx path only (i.e. for ATIS transmissions), especially in the case that a transmitter doesn't have an associated receiver.

Note 9.

It should be noted that recording audio on the Receive path only from a Radio Receiver will also include transmitted audio due to the radio receiver(s) picking-up the off-air transmitted audio from its associated Radio Transmitter. Transmitted audio recorded on the receive path is also proof that the audio was actually transmitted. In the case of a Radio Transceiver, summation of incoming and outgoing audio streams at the VCS client prior to sending a single audio stream to the RTSP server (Recorder equipment) MAY be necessary.

3.9 Recording of Air-Ground audio communications at the Radio/Radio Gateway endpoint

In the case of client being located at the Radio/Radio gateway endpoint, a Radio or a Radio Gateway network interface client (connecting legacy radio to an IP network) with an established A/G communication **SHOULD** generate a single RTP audio stream that **MAY** contain one of the following:

- The incoming (Rx) audio stream only, in the case of a Radio receiver (optional)
- The sum of the outgoing (Tx) audio stream and the incoming (Rx) audio stream, in the case of a Radio transceiver. (optional)

The Radio/Radio Gateway **SHOULD** establish a single RTSP session with its nominated RTSP server and **MAY** establish dual RTSP sessions with its nominated RTSP server for redundancy purposes.

Note 10.

Recording at a Radio/Radio gateway network interface client does confirm that controller speech was received from the network at the radio transmitter, but isn't confirmation that speech was actually transmitted. It also confirms that audio was received from a radio receiver prior to being sent over the network, but doesn't confirm that the controller listened to the audio recorded at this recording point. Recording at this point can be useful to demonstrate in case of an audio path failure over the network for example or a Transmitter/Receiver failure.

3.10 Provision for 2 identical autonomous voice recordings at VCS endpoint

In the case of recording G/G or A/G related communications at the VCS endpoint, the Voice Recording system **SHALL** provide a means by which two autonomous audio recordings can be made. Both recordings **SHALL** be identical in all respects.

Note 11.

This requirement relates to two independent RTSP servers (Recorder equipment) being used to record audio at the same recording point and also to increase the availability of recorders in case of failure, maintenance etc. It should be noted that the audio recording relating to G/G communications and A/G communications shall occur independently at each endpoint involved in the call.

3.11 Provision for dual channels for position recording

Each position **SHALL** support up to two voice recording channels.

The specific selection of the audio sources and their assignment to the two voice recording channels will be ANSP-specific.

In the case of recording G/G or A/G related communications at the Position, the Voice Recording system **SHALL** provide a means by which audio and Call Record Data messages are sent to the RTSP server (Recording equipment) over two independent RTSP sessions in order to achieve redundancy. Both audio streams **SHALL** be identical.

3.12 Provision for at least one voice recording at Radio or Radio Gateway endpoint

In the case of recording A/G communications at the radio or radio gateway endpoint, the Voice Recording system **SHOULD** provide a means by which at least one audio recording can be made and **MAY** provide a means by which two autonomous audio recordings can be made. In the case that two audio recordings are made, then both recordings **SHALL** be identical in all respects.

Note 12.

This requirement relates to two independent RTSP servers being used to record audio at the same recording point and also to increase the availability of recorders in case of failure, maintenance etc. It should be noted that in the case of Remote Radio sites with limited bandwidth capabilities, it is not mandatory to record audio directly at the radio/radio gateway. The A/G audio shall always been recorded at the VCS endpoint.

3.13 True and faithful representation of audio signal

The Voice Recording system **SHALL**, at all times, provide a means for recording voice which is a true and faithful representation of the audio signals being presented at the points detailed in requirement 3.2 “Voice recording entities” 3.3 “Voice location identification” above.

The quality of the audio being recorded at the RTSP server (Recorder equipment) **SHALL** be identical to that at the point of where it was recorded.

From the recording point to the RTSP server, the audio quality **SHALL NOT** be degraded or improved.

The RTSP server **SHALL NOT** degrade or improve the audio quality during the recording of the audio.

The RTSP server **SHALL NOT** regenerate lost packets using post-processing techniques.

The RTSP server **SHALL NOT** perform transcoding of the received audio signal. It **SHALL** store the audio using same audio codec defined during RTSP session establishment.

Note 13.

Audio Recording at the User Terminal client should record audio as close as possible to the controller earpiece or loudspeaker and as close as possible to the Controller microphone.

Audio Recording at the VCS or Radio network interface client should record the audio signal as it appears at that point.

Should a Speech decoder have Packet Loss Concealment (PLC) algorithm enabled for example leading to the automatic regeneration of lost packets at the received side based on pre and post speech spectral analysis (or another speech enhancement algorithm active), the received audio stream being sent to the RTSP server should occur after any enhancement process has been applied at the receive side in order that the audio heard by the controller is identical to that being sent towards the RTSP server.

3.14 Date & Timestamp Synchronization for audio

For the purposes of imprinting date and timestamp information with the recorded audio (in order to enable the time and date of re-played voice messages to be precisely identified), the Voice Recording system **SHALL** be synchronous with the Universal Time Coordinated (UTC) date and time data source to the accuracy specified by the ICAO Convention on International Civil Aviation, Annex 5: “Units of Measurement to be Used in Air and Ground Operations” [34].

- YYYY-MM-DD_HH:MM:SS.XXX+0000

The date **SHALL** be indicated in numeric form: “YYYY-MM-DD”

The time **SHALL** be indicated in numeric form as “HH:MM:SS”

The timestamp **SHALL** have a resolution of 1 millisecond.

3.15 Date & Timestamp Synchronization for Call Record Data

For the purposes of imprinting date and timestamp information with the Call Record Data (in order to enable the time and date of call record events to be precisely identified), the Voice Recording system **SHALL** be synchronous with the Universal Time Coordinated (UTC) date and time data source to the accuracy specified by ICAO Convention on International Civil Aviation, Annex 5: “Units of Measurement to be Used in Air and Ground Operations” [34].

Call Record Data (CRD) relating to telephone calls **SHALL** be sent at call initiation, call accept/refusal and at call termination.

Call Record Data (CRD) relating to outgoing radio calls **SHALL** be sent at PTT activation and deactivation. Call Record Data (CRD) relating to incoming aircraft calls **SHALL** be sent at SQU activation and SQU deactivation.

The timestamp **SHALL** be set by the client because it has the exact time reference for any local event.

It **SHALL** be possible for a client to use a local time set and convert to a UTC timestamp when sending CRD to the RTSP server.

If a timestamp value is omitted by the client (as could happen in the case of Remote Radios within connection to an NTP server), the RTSP server **SHALL** assign a timestamp relative to message arrival at the RTSP server.

3.16 Voice Packet size towards RTSP server

The RTSP server (Recorder equipment) **SHALL** have a capability to recognize and record audio packets having a size from 10ms to 80ms. Packet size negotiation during the session establishment with the RTSP server **SHALL NOT** occur.

Although the default audio packet size towards the RTSP server **SHALL** be 20ms, it **SHOULD** be possible for the client to use larger packet sizes in order to reduce bandwidth overhead.

3.17 Diffserv codepoint for Recorder audio

The DiffServ codepoint assigned to IP packets being sent to an RTSP server **SHALL** be configurable but have a setting of at least AF31 or better.

3.18 Audio codecs

The Terminal/VCS/Radio clients and RTSP server **SHALL** have the following audio codecs available:

- ITU-T Rec. G.711 PCM A-law (in Europe)
- ITU-T Rec. G.711 PCM mu-law (in North America/Japan)

The G.711 PCM mu-law codec **SHOULD** be used as default audio codec by Terminal/VCS/Radio clients and RTSP server (Recorder equipment).

The Terminal/VCS/Radio clients and RTSP server **MAY** have the following audio codecs available:

- ITU-T Rec. G.728 LD-CELP (Compressed 16kbps without overhead)
- ITU-T Rec. G.729 CS-ACELP (Compressed 8kbps without overhead)

3.18.1 Recording compressed audio streams

When an audio stream at a VCS network interface has been compressed prior to its transport over the network (for reduced bandwidth consumption reasons), the audio **SHALL** be sent to an RTSP server in an identical compressed format. The same audio codec **SHALL** be defined when the RTSP session is opened to the RTSP server.

The RTSP server in this case **SHALL NOT** perform any transcoding of the audio signal prior to its storage.

3.18.2 Recording uncompressed audio streams

When audio stream received at a position has been decompressed after its transport over the network (for reduced bandwidth consumption reasons), the audio **SHALL** be sent to the RTSP server at the position in an identical uncompressed format (i.e. G.711 mu or A law). The same audio codec **SHALL** be defined when the RTSP session is opened to the RTSP server.

The RTSP server in this case **SHALL NOT** perform any transcoding of the audio signal prior to its storage.

Note 14.

The summing of audio streams is easier to perform if they all have a common uncompressed format (i.e. G.711).

3.18.3 Audio codec request to RTSP server

The client **SHALL** be able to request the use of a codec during the RTSP session establishment with an RTSP server.

The RTSP server **SHALL** be able to reject an RTSP session request if it does not support the requested codec.

3.18.4 Audio encryption over RTSP session

Audio being sent to the RTSP server by the client **MAY** be encrypted using strong encryption technology like 128Bit AES encoding. In the case that encryption is used, stored voice samples **SHALL** be encoded by 128/256 Bit AES algorithm.

Note 15.

It is recommended that audio recorded by the RTSP server is stored in an encrypted format, because it ensures that the recorded audio files can't be tampered with.

3.19 SIP

3.19.1 SIP session establishment with Recorder

The use of SIP to exchange capabilities and connection information (i.e. IP address, port number, and transport protocol) between VCS/Radio client and RTSP server involved in a recording session **MAY** be employed (i.e. the use of the SIP protocol is optional). The opening of a permanent SIP session between VCS/Radio client and RTSP server involved in a recording session is also optional.

Note 16.

In the case that SIP is not employed, it is necessary to define statically configured recorder binding [IP:Port] in order to employ RTSP implementations;

3.19.2 Registration with SIP Registrar Server

In the case that SIP is employed, any VCS/Radio client or RTSP server (Recorder equipment) involved in a recording session **SHOULD** register with a SIP Registrar server using the REGISTER method according to RFC3261 [7]. It **SHOULD** be possible to register multiple contacts for a single Address of Record (AOR).

3.19.3 Transport protocol

In the case that SIP is used, the SIP session description **SHALL NOT** specify the transport protocol to be used.

3.20 Transport protocol

The RTSP session description **SHALL** specify the transport protocol to be used.

Transport of media **SHALL** be based on RTP over independent TCP (default) and **MAY** be based on RTP over UDP.

Note 17.

RTP over independent TCP is also known as non-interleaved binary data.

Note 18.

The choice of RTP over independent TCP implies that RTSP signaling and RTP media shall be split on different sockets. A split will allow greater flexibility in the future, as RTP is sent over TCP and could eventually be sent over UDP (only if proven to be reliable), while RTSP is sent over TCP and could also eventually be sent over UDP (only if proven to be reliable).

Transport of media **SHALL NOT** be based on Embedded (interleaved) Binary Data.

3.21 RTP

3.21.1 Transport of audio

RTP over TCP **SHALL** be used for the transport of audio (default). TCP provides a guaranteed delivery service for the audio packets towards the RTSP server (recorder equipment).

RTP over UDP **MAY** be used to transport packets towards the RTSP server (recorder equipment).

Note 19.

Although audio for G/G and A/G communications will use UDP for the transport of real time media over a network, the time stamped audio packets sent to an RTSP server will use TCP in order to ensure that every audio packet is recorded. Time stamped audio packets being sent to an RTSP server therefore don't have to be sent in real time over the network (in the case of a lost packet being sent again for example), but shall be recorded by the RTSP server with their original timestamp in order that any playbacks will result in the original timestamps being applied.

Note 20.

It is recommended that the timestamps being used in the RTP timestamp field are incremented relative to the number of voice samples present in the voice packet with respect to the codec being used. For example in the case of a G.711 mu-law codec, the number of voice samples in a 20ms voice packet will be 160, while a 30ms voice packet will contain 240.

3.21.2 Transport of RTP header extension

A VCS/Radio client **MAY** include the RTP Header Extension in an RTP stream being sent to an RTSP server (Recording equipment).

An RTSP server (Recording equipment) **SHALL** have the capability of recording audio from an RTP stream independently of there being an RTP header extension being present.

The RTSP server **MAY** have the capability of processing the RTP Header extension from streams in which it is present.

3.22 RTP with RTCP

It **SHALL** be possible for a VCS/Radio client to use either RTP only or RTP with RTCP.

Note 21.

As the RTP stream is one-way towards the RTSP server and TCP is being used for the transport protocol, the advantages that an RTCP implementation can provide over a unidirectional RTP path are considered minimal.

3.23 RTSP

3.23.1 Transport of messages

VCS/Radio Clients and RTSP Servers involved in a recording **SHALL** have the capability to support RTSP over TCP (default setting) and **MAY** have the capability to support RTSP over UDP.

Note 22.

RFC2326 [5] defines RTSP over TCP as mandatory and RTSP over UDP as optional.

3.23.2 RTSP server port

The default port for the RTSP server **SHALL** be 554 for both TCP and UDP (if employed).

Note 23.

A VCS or Radio client will therefore establish a TCP connection to the RTSP server (Recorder equipment) on TCP port 554, the main port for RTSP.

3.23.3 RTSP addresses

Each RTSP server (Recorder equipment) **SHALL** have the capability of being assigned a unique RTSP address such that RTSP sessions can be established to it from the VCS client or from the Radio client.

Note 24.

An example of an RTSP address is defined as follows:

- rtsp://ANSP-recorder2:554/iprecorder/ RTSP/1.0

3.23.4 RTSP address permissions list

Each VCS and Radio client **SHALL** have a RTSP address permissions list, defining the RTSP servers (Recorder equipment) to which it is allowed to establish RTSP sessions.

Each RTSP server (Recorder equipment) **SHALL** have a RTSP address permissions list, defining the User Terminal, VCS and Radio clients to which it is allowed to accept RTSP sessions requests.

Terminal/VCS/Radio clients and RTSP server (Recorder equipment) **SHALL** refuse RTSP session establishment attempts to/from RTSP addresses not defined in the permissions list.

It **SHALL** be possible to disable the RTSP address permissions list check at each VCS/Radio client and RTSP server. In this case the VCS/Radio clients and RTSP server (Recorder equipment) **SHALL** accept RTSP session establishment attempts to/from any RTSP address.

3.23.5 OPTIONS request message

RTSP servers **SHALL** support the OPTIONS request message.

An OPTIONS request message sent by a User Agent client to the RTSP server **SHALL** return all the request message types that the RTSP server will accept.

3.23.6 RTSP session establishment for G/G communications

Dual RTSP sessions **SHALL** therefore be established per incoming or outgoing G/G communication by each User Terminal client or VCS client to its nominated RTSP server (Recorder equipment).

The audio relative to one G/G communication between two parties **SHALL** therefore result in both clients establishing **dual RTSP sessions** with its nominated RTSP server and sending its summed audio stream.

In the case of a single G/G communication (i.e. one DA call, IA or OVR call only) between two endpoints, each endpoint **SHALL** establish **dual RTSP sessions** with its RTSP server for transfer of CRD and summed audio.

In the case of multiple G/G communications at one endpoint within the system (i.e. Outgoing DA call and incoming OVR call), the endpoint **SHALL** establish **dual RTSP sessions** to its RTSP server for the first G/G communication in order to transfer the sum of the incoming and outgoing audio, but it **SHALL NOT** establish a new RTSP session for other G/G communications occurring concurrently with the first. Each new G/G communication **SHALL** have its incoming and outgoing audio summed with that of the previously established G/G communications (as if in a conference) towards the RTSP server **over both RTSP sessions**. CRD is generated per individual G/G communication.

Note 25.

A position receiving an OVR call for example will establish dual RTSP sessions with its RTSP server in order to transfer the summed audio stream and send CRD related to this call. Should the position now establish an outgoing DA call, it will not be necessary to establish a new session. The CRD related to the DA call should be sent. It will be necessary to only add the incoming audio for the DA call to the summed audio stream being sent to the RTSP server because the position microphone was already being sent before.

Also in the case of a conference call at a position with calls being made to several other positions, each position will establish **dual RTSP sessions** to its RTSP server. Incoming audio on all conference connections will be summed with local microphone outgoing audio and sent to the RTSP server.

3.23.7 RTSP session termination for G/G communications

Termination of the G/G communication **SHALL** cause a teardown of the **dual RTSP sessions** by the User Terminal/VCS client towards its nominated RTSP server.

In the case of Call intrusion, Call Conference or call combinations (i.e. DA call with incoming IA call), the **dual RTSP session** termination **SHALL** occur on termination of the last remaining call.

3.23.8 RTSP session establishment for A/G communications

In the case of A/G communications at a position, **dual RTSP sessions** **SHALL** be established from the User terminal client to the RTSP server (Recorder equipment). This occurs when at least one radio frequency is configured within the position.

In the case of further frequencies being configured within the position another RTSP session **SHALL NOT** be established from the User terminal client to the RTSP server (Recorder equipment).

In the case of multiple frequencies being configured within the position, summed transmit and receive audio from all the frequencies **SHALL** be sent on the **dual RTSP sessions** to the RTSP server.

Note 26.

The RTP stream sent to RTSP server is the audio sum of all frequencies at the position and relates to what the controller actually hears and speaks.

In the case of frequencies enabled for Receiver voting configured within the position (i.e. multiple receivers operating on same frequency), summed audio from the voted receivers only plus all other frequencies (without Receiver voting) **SHALL** be sent on the **dual RTSP sessions** to the RTSP server.

In the case of Main and Standby radios for each frequency being configured within the position, summed audio from Main and Standby radios **SHALL** be sent on the **dual RTSP sessions** to the RTSP server.

Note 27.

Either the Main or Standby radio would be selected at the position at any instant.

In the case of A/G communications at a VCS client, one RTSP session **SHALL** be established from the VCS client to the RTSP server (Recorder equipment) per radio frequency. **Dual RTSP sessions MAY be established.** In the case of multiple frequencies being configured within the system, one RTSP session **SHALL** be established to the RTSP server for each frequency. **Dual RTSP sessions MAY be established.**

In the case of Main and Standby radios existing per frequency, one RTSP session **SHALL** be established from the VCS client to the RTSP server (Recorder equipment) per radio frequency. Summed audio from Main and Standby radios **SHALL** be sent on the single RTSP session to the RTSP server. **Dual RTSP sessions MAY be established.**

In the case of A/G communications at a Radio client, one RTSP session **SHOULD** be established from the Radio (Tx, Rx or TxRx) to the RTSP server (Recorder equipment) relative to the radio frequency, when a SIP session request from the VCS has been accepted by the radio. **Dual RTSP sessions MAY be established.**

In the case of Main and Standby radios existing per frequency, one RTSP session **SHOULD** be established from the Main Radio (Tx, Rx or TxRx) to the RTSP server (Recorder equipment) and one RTSP session **SHOULD** be established from the Standby Radio (Tx, Rx or TxRx) to the RTSP server (Recorder equipment). **Dual RTSP sessions MAY be established.**

Note 28.

As Main or Standby radios are physically separated, summed audio from Main and Standby radios can't be sent on a single RTSP session to the RTSP server. The RTSP

server would only receive audio streams from one radio at a time. The RTSP server should have capability to associate two RTSP sessions with one frequency.

The RTSP sessions established at both VCS and Radio clients with its nominated RTSP server (Recorder equipment) **SHALL** be independent.

3.23.9 RTSP session terminated for A/G communications

In the case of A/G communications at a position, the **dual RTSP sessions SHALL** remain established from the User terminal client to the RTSP server (Recorder equipment) while at least one radio frequency is configured within the position. If the last frequency is removed from position, the **dual RTSP sessions** with the RTSP server **SHALL** be torn down.

In the case of A/G communications at a VCS client, the RTSP session(s) **SHALL** remain established from the VCS client to the RTSP server (Recorder equipment) while the SIP session remains established between VCS and radio. On SIP session termination (either from the VCS or radio endpoint), the RTSP session(s) with the RTSP server **SHALL** be torn down.

In the case of A/G communications at a Radio client, the RTSP session(s) **SHALL** remain established from the Radio client to the RTSP server (Recorder equipment) while the SIP session remains established between VCS and radio. On SIP session termination (either from the VCS or radio endpoint), the RTSP session(s) with the RTSP server **SHALL** be torn down (if existing).

3.23.10 Lost audio prevention during RTSP session

The RTSP server **SHALL** have a mechanism to detect and report missing RTP packets from an RTP stream received from User Terminal/VCS/Radio client. Due to RTP being sent over TCP, audio packets lost during transport to the RTSP server **SHALL** be re-sent by the VCS/Radio client, due to packets remaining in the retransmission buffer until acknowledged.

3.23.11 Lost audio prevention during RTSP session teardown

The RTSP server **SHALL** ensure that in the case of RTP audio packets being received on an established session with a client and a TEARDOWN message to closed the session is also received from the same client, it **SHALL NOT** close the session until audio packets have stopped arriving. It is therefore necessary to wait until the client has closed the session first and has therefore stopped sending audio packets.

3.24 Bandwidth to RTSP server

Allocated bandwidth between a User Terminal/VCS/Radio client and RTSP server for a single RTP session **SHALL** be sufficient to ensure that congestion does not occur and packets are not dropped.

Allocated bandwidth between a User Terminal/VCS/Radio client and RTSP server **SHALL** be sufficient to ensure that a drop in throughput does not occur due to the re-transmission of lost packets.

3.24.1 RTSP session timeout

An RTSP session **SHALL** have a means to assure link connectivity is active between the client and RTSP server (i.e. enabling the use of a TCP Keepalive mechanism).

On loss of an RTSP session, mechanisms **SHALL** be in place for its immediate re-establishment.

3.25 RTSP server session limit

An RTSP server **SHALL** have the capability of handling multiple sessions simultaneously.

An RTSP server **SHALL** have the capability of handling a quantity of simultaneous sessions proportional to system size and the busy hour peak traffic calculated for the system. Ideally the RTSP server **SHALL** be scalable over time, such that it has extra capacity in order to cater for future equipment expansions (i.e. extra positions) and higher number of RTSP sessions.

Note 29.

For example the RTSP server in an ATS unit with 20 positions, would be expected to handle at least 40 concurrent sessions (G/G + A/G) excluding any future scalability.

3.26 Client session limit

A User Terminal client **SHALL** have the capability of handling multiple sessions simultaneously. This **SHALL** include at least one session for G/G communication, one session for all current A/G communications.

3.27 Referencing G/G communications

The following reference values **SHALL** be generated in order to identify G/G communications.

- RTSP Session id: a unique session identity generated by an RTSP server (Recorder equipment). The session identifier is communicated to the VCS client during RTSP session establishment.
- connref: a unique local connection reference generated by a VCS client endpoint for each RTSP session with its nominated RTSP server. One connref value is generated per G/G communication and used throughout lifetime of the call. Each VCS client involved in call would generate a different local connref value towards its RTSP server for the same call.
- CallRef: generated by a VCS client on initiating a communication and used by all VCS clients involved in the communication as a common reference to identify the communication. Each VCS client involved in call would therefore use the same CallRef towards its RTSP server in order to identify the call.
- ThreadRef: generated by a VCS client on initiating a communication and used by all VCS clients involved in the communication as a reference that identifies a sequence of operations relating to the call. If a call is transferred for example, a new CallRef is generated at the instant of the transfer, but the Thread reference remains the same. Each VCS client involved in call would therefore use the same ThreadRef towards its RTSP server in order to identify the operations related to a call.

3.27.1.1 RTSP Session Identity Reference value

A RTSP server (Recorder equipment) **SHALL** be responsible for generating a unique Session identifier following a RTSP session establishment request from the VCS client. The session identifier **SHALL** be at least eight octets long.

3.27.1.2 Connection Reference value

A VCS client endpoint **SHALL** be responsible for generating a unique local Connection reference value for each RTSP session towards its nominated RTSP server. One connref **SHALL** be generated per RTSP session and is therefore relevant to one G/G communication. The messages sent by a client to its RTSP server related to the call will contain the same connref value. Each VCS client involved in call would generate different local connref towards its RTSP server for the same call.

3.27.1.3 Call Reference value

It is necessary that all clients involved in a G/G communication are able to have a common call reference assigned to the communication in order that it can be identified at the RTSP server.

The VCS client endpoint initiating the call request **SHALL** generate a Call Reference value. The VCS client endpoint receiving the call request **SHALL** use the same call reference to assign its recordings and identify the communication.

Each VCS client endpoint **SHALL** then employ this common Call Reference value when establishing an RTSP session to its nominated RTSP server. This method **SHALL** allow audio recordings for a call to be identified in different RTSP servers. The call reference value **SHALL** also be used by clients when sending the Call Record Data to the RTSP server.

3.27.1.4 Thread Reference value

It is necessary that all clients involved in a G/G communication are able to have a common Thread reference assigned to the communication that identifies a sequence of operations relating to the call throughout its lifetime and which can be used by the RTSP server to identify these operations.

A VCS client endpoint initiating the call request **SHALL** generate a Thread Reference value. This **SHALL** be defined as a SIP header. The VCS client endpoint receiving the call request **SHALL** use the same Thread reference to assign its recordings and identify the communication.

Note 30.

If a DA call is transferred for example, a different CallRef is generated for each DA call, but the ThreadRef remains the same.

An example of a Call Transfer is when "A" calls "B", "B" places "A" on hold and then calls "C", "B" then transfers "A" to "C" by sending REFER to A (attended or unattended call transfer) which will trigger "A" to establish call to "C". With respect to ThreadRef generation the following occurs:

- "A" calls "B" and generates ThreadRef1;*
- "B" calls "C" and generates ThreadRef2;*
- "B" then transfers "A" to "C" by sending a REFER to A*
- "A" will then call "C" using its original ThreadRef1;*

In the above example, the ThreadRef included in CRD sent by "A" to its RTSP server would be the same for the call from "A" to "B" and for the call from "A" to "C". This allows operations effected during the call lifetime to be identified.

Note 31.

In the case of a node receiving multiple incoming OVR calls for example, each call would have its own unique ThreadRef. If the node now generates its own outgoing OVR call, a new ThreadRef will be generated as each OVR call is treated independently. It should also be noted that it is forbidden to transfer, place on-hold or intrude into an OVR call.

3.28 Referencing A/G communications

The following reference values **SHALL** be generated in order to identify A/G communications.

- RTSP Session id: a unique session identity generated by an RTSP server (Recorder equipment). The session identifier is communicated to the VCS/Radio client during RTSP session establishment.
- connref: a unique reference generated by a VCS/Radio client endpoint for each PTT/SQU activation. One connref **SHALL** be generated per PTT/SQU activation and is therefore relevant to one A/G communication. The connref will therefore allow identification of audio messages at the RTSP server.
- ThreadRef: generated by a VCS client on initiating a SIP session and used by all VCS and Radio clients involved in the SIP session as a reference that identifies a sequence of operations relating to the session. Each VCS and Radio client involved in SIP session would therefore use the same ThreadRef towards its RTSP server in order to identify the operations related to a session.

3.28.1.1 RTSP Session Identity Reference value

A RTSP server (Recorder equipment) **SHALL** be responsible for generating a unique Session identifier reference following a RTSP session establishment request from the VCS or Radio client. The session identifier **SHALL** be at least eight octets long.

3.28.1.2 Connection Reference value

A VCS/Radio client endpoint **SHALL** be responsible for generating a unique Connection reference value for each PTT/SQU activation. A PTT activation and deactivation event **SHALL** have the same connref. A SQU activation and deactivation event **SHALL** have the same connref. The connref will therefore allow identification of audio messages at the RTSP server. Both VCS and Radio client endpoints would therefore generate different connref values towards its nominated RTSP server for the same audio message. Identification of the same message at different RTSP servers can only occur through UTC time stamps.

3.28.1.3 Thread Reference value

It is necessary that all clients involved in an A/G communication are able to have a common Thread reference assigned to a SIP session that identifies a sequence of operations relating to the session throughout its lifetime and which can be used by the RTSP server to identify these operations.

A VCS client endpoint initiating the session request **SHALL** generate a Thread Reference value. The VCS client endpoint receiving the session request **SHALL** use the same Thread reference to assign its recordings and identify the session.

Note 32.

If a SIP session is switched from a Main to a Standby radio for example, a new CallRef is applied to the SIP session with the Standby radio, but the Thread reference remains the same as that used to the Main Radio. Each VCS and Radio client involved in SIP session would therefore use the same ThreadRef towards its RTSP server in order to identify the operations related to a session.

3.29 Call Record Data for G/G communications

Call Record Data (CRD) relating to G/G communications **SHALL** normally be sent at call initiation, call accept/refusal and at call termination.

It **SHALL** be possible for a client to send CRD at any time to the RTSP server even without an RTSP session established (hence no session identifier assigned). This will simply add information and **SHALL NOT** initiate or teardown a session.

This **SHALL** allow the following:

- to record CRD of calls that have not been successfully set up (ex. gateway is blocked, congestion, access list reject)
- to record CRD from the client without having to wait for a session setup to complete
- after the call has been cleared and the corresponding RTSP session has been torn down, it will be possible to add extra information about the call (ex. comments, ratings etc.).

3.29.1 Call Record Data Properties for G/G communications

The Call Record Data properties relating to G/G communications **SHALL** include the following:

Property	Mandatory (M) / Optional (O)
Direction	M
Priority	M
CallingNr	M
CalledNr	M
AlertingNr	O
ConnectedNr	O
SetupTime	M
AlertTime	O
ConnectTime	O
DisconnectTime	M
DisconnectCause	M
DisconnectSource	M
Call Type	M
CallRef	M
ThreadRef	O

3.29.2 Call Record Data Operations for G/G communications

The Call Record Data operations relating to G/G communications **SHALL** include the following:

Operation	Mandatory (M) / Optional (O)
RedirectedNr	M
TransferredNr	O
Hold	O

3.29.3 Call intrusion and conference call management towards RTSP server

The VCS client **SHALL** have the capability to manage audio and Call Record Data for Call intrusion and Conference calls.

The following sequence of events **SHALL** be followed for a call intrusion or conference call at a position.

1. On establishment of first call, the VCS client establishes RTSP session(s) with RTSP server:
2. VCS client sends CRD to inform RTSP server about presence of first call.
3. VCS client sends summed audio (over dual RTSP sessions) from first call to RTSP server.
4. On intrusion or connection of second call, VCS client sends CRD to inform RTSP server about presence of second call.
5. VCS client now sends summed Audio (over dual RTSP sessions) from all parties to RTSP server.
6. In case of conference call, further parties can be added using same procedure.
7. On first or second call being cleared, the VCS client sends CRD to inform RTSP server about the clearing of call.
8. VCS client now sends summed audio (over dual RTSP sessions) from remaining connected parties to RTSP server.
9. When last call is cleared, VCS client sends CRD to inform RTSP about the clearing of last call.
10. VCS client now clears RTSP session and audio therefore no-longer sent to RTSP server.

3.29.4 Call Hold management towards RTSP server

The VCS client **MAY** have the capability to manage a Call Hold event to the RTSP server. In the case that the Call Hold feature is implemented it **SHALL** indicate to the RTSP server when a call is put on hold and by which party (calling or called). It **SHALL** indicate to the RTSP server when a call is removed from hold

3.30 Call Record Data for A/G communications

Call Record Data (CRD) relating to A/G communications **SHALL** be sent to the RTSP server each time there is a change in control bits being sent to the Radio or Radio Remote Control Equipment (RRCE) and when there is a change in the status indication bits being received from the Radio or RRCE.

CRD **SHALL** therefore be sent in the following circumstances:

1. When there is a change in PTT-type control bits being sent towards a Radio or RRCE;
2. When there is a change in PTT-Type status bits being received as acknowledgement from a Radio or RRCE;
3. When there is a change in the SQU status bit being received from a Radio or RRCE (configured in single frequency mode);
4. When there is a change in PTT summation bit (PTTS) being sent to or received from a Radio or RRCE;
5. When there is a change in PTT Mute bit (PM) being sent to or received from a Radio or RRCE;
6. When there is a change in Simultaneous Transmission Bit (SCT) being received from a Radio;

In order for RTSP server to have sufficient information in order to determine if a PTT activation resulted in a transmission, the CRD **SHALL** associate each sent PTT-Type or acknowledged PTT-type with its PTTid.

Note 33.

If an acknowledgement from a radio or RRCE has a PTT-type and/or a PTTid that differs from the PTT-type command sent to the radio or RRCE, post processing of CRD at the RTSP server would indicate that the PTT activation attempt had been unsuccessful (i.e. due to a radio or RRCE in lock-out mode with a transmission already in progress).

In the case of a Radio Remote Control equipment (RRCE) (configured for either single frequency or paired frequency mode), the Call Record Data (CRD) relating to A/G communications **SHALL** also be sent to the RTSP server each time a new an RRC command message is sent to the RRCE and each time an RRC response message is received from the RRCE. As several consecutive RRC command messages could be sent, each containing the same information, it would be necessary to send the CRD only once for the series of identical RRC control messages.

Call Record Data sent to the RTSP server for a RRC command message (RRCC) and a RRC response message (RRCR) **SHALL** indicate the eight RRC control bits. These bits include:

1. MSTxF1 - Main/Standby Transmitter F1
2. MSRxF1 - Main/Standby Receiver F1
3. MSTxF2 - Main/Standby Transmitter F2
4. MSRxF2 - Main/Standby Receiver F2
5. SelTxF1 – Select Transmitter F1
6. SelTxF2 – Select Transmitter F2
7. MuRxF1 – Mute Receiver F1
8. MuRxF2- Mute Receiver F2

In the case of a Radio Remote Control equipment (RRCE) configured in paired frequency mode, the Call Record Data (CRD) relating to A/G communications **SHALL** be sent to the RTSP server each time a squelch activation/deactivation event occurs on frequency f1 (SQF1) or on frequency f2 (SQF2). Call Record Data **SHALL** indicate the state of the SQF1 and SQF2 status being received from the RRC and **MAY** indicate relative Signal quality information for each signal if Receiver Voting is enabled. These bits include :

1. SQF1 – Squelch on Frequency F1
2. SQF2 – Squelch on Frequency F2
3. SQI F1- Signal Quality Information of Receiver F1 (optional)
4. SQI F2- Signal Quality Information of Receiver F2 (optional)

It **SHALL** also be possible for a client to send CRD at any time to the RTSP server even without an RTSP session established (hence no session identifier assigned). This will simply add information and **SHALL NOT** initiate or teardown a session.

This **SHALL** allow the following:

- to record CRD of radio sessions that have not been successfully set up (ex. gateway is blocked, congestion, access list reject);
- to record CRD from the client without having to wait for a session setup to complete;
- after the radio session has been cleared to a radio and the corresponding RTSP session has been torn down, it will be possible to add extra information about the session (ex. comments, ratings etc.).

3.30.1 Call Record Data Properties for A/G communications

The Call Record Data properties relating to A/G communications **SHALL** include the following:

Property	Mandatory (M) /Optional (O)
FrequencyID	M
PTTid	M
BSS Method	O
Calling Address	M
Called Address	M
Direction	M

3.30.1.1 Call Record Data Operations for A/G communications

The Call Record Data operations relating to A/G communications **SHALL** include the following:

Operation	Mandatory (M) / Optional (O)
PTT (OFF, Normal, Coupling, Priority, Emergency)	M
SQUELCH	M
PTTS (PTT summation)	M
PM (PTT Mute)	M
Simultaneous Transmission	M
Receiver Voting (Best Signal Selection)	O
BSS Quality Index	O (Radio or RRCE in single frequency mode only)

In addition the Call Record Data operations relating to A/G communications for Radio Remote Control Equipment (RRCE) configured in Single or Paired frequency mode **SHALL** include the following:

Operation	Mandatory (M) / Optional (O)
RRCC (command)	M
RRCR (response)	M

In addition the Call Record Data operations relating to A/G communications for Radio Remote Control Equipment (RRCE) in Paired frequency mode only **SHALL** include the following:

Operation	Mandatory (M) / Optional (O)
SQF1	M
SQF2	M
BSS Quality Index (SQI F1)	O
BSS Quality Index (SQI F2)	O

3.30.1.2 PTT operation for A/G communications

The Call Record Data indicating a PTT operation that is sent to the RTSP server **SHALL** identify PTT-type as one of the following values: PTT OFF, Normal PTT ON, Coupling PTT ON, Priority PTT ON or Emergency PTT ON.

This information is sent together with PTTid each time there is a change in state of the PTT-type being sent to a radio and when there is a change in the PTT-type acknowledgement status being received from a radio.

3.30.1.3 SQU operation for A/G communications

The Call Record Data indicating a SQU operation that is sent to the RTSP server (in the case of Radio Rx, TxRx or RRCE operating in single frequency mode), **SHALL** identify SQU-type as one of the following values: SQU OFF or SQU ON.

In the case of RRCE (operating in Paired frequency mode), the Call Record Data indicating a SQU operation that is sent to the RTSP server **SHALL** identify SQU-type as one of the following values

1. SQF1 ON or OFF
2. SQF2 ON or OFF

3.30.2 Management of Simultaneous aircraft calls on different frequencies towards RTSP server

The User Terminal client **SHALL** have the capability to manage audio and Call Record Data to the RTSP server relating to multiple aircraft calls occurring simultaneously on different frequencies.

The following sequence of events **SHALL** be followed for simultaneous aircraft calls occurring on different frequencies at a position with a RTSP session already established to a RTSP server.

1. On arrival of the first SQU ON signal, the client sends CRD to inform RTSP server about the presence of first aircraft call.
2. Client sends audio from first call (over dual RTSP sessions) to RTSP server.
3. On arrival of the second SQU ON signal, the client sends CRD to inform RTSP server about the presence of second aircraft call.
4. Client sends summed audio from both aircraft calls (over dual RTSP sessions) to RTSP server.
5. On arrival of the first SQU OFF signal, the client sends CRD to inform RTSP server about the presence of one aircraft call only.
6. Client sends audio from remaining aircraft calls (over dual RTSP sessions) to RTSP server
7. On arrival of the second SQU OFF signal, the client sends CRD to inform RTSP server about the presence of no aircraft calls.
8. Client no longer sends audio to RTSP server

3.30.3 Management of Receiver Voting operation towards RTSP server

The User Terminal client **SHOULD** have the capability to manage audio and Call Record Data to the RTSP server relating to Receiver Voting (i.e. Best Signal Selection).

The User Terminal client **SHOULD** inform the RTSP server about which Rx stream is selected by the voting algorithm when multiple streams are being received from different Radio Receivers.

The User Terminal client **SHOULD** also inform the RTSP server about which Rx stream are not selected by the voting algorithm when multiple streams are being received from different Radio Receivers.

3.30.3.1 Receiver voting for A/G communications

The Call Record Data indicating a VOTING operation that is sent to the RTSP server **SHALL** identify VOTING status of the Squelch signal as one of the following values: Voting disabled, Selected as best voted, Not selected as best voted.

3.31 Replay

It **SHOULD** be possible for each User terminal to establish a Replay session.

Each RTSP server (Recorder equipment) **SHOULD** have RTSP replay functionality (i.e. Replay client) within the Recorder equipment.

The Replay client within the RTSP server (Recorder equipment) **SHOULD** have capability to receive the DESCRIBE, SETUP, PLAY, PAUSE and TEARDOWN messages.

The Replay client within the RTSP server **SHOULD** be able to identify recorded audio to be played back when receiving a DESCRIBE message.

Individual audio messages **SHALL** be identified through the connref.

On receiving a PLAY message, the Replay client within the RTSP server **SHOULD** be able to send an audio stream to the requestor that contains the audio identified by the DESCRIBE message.

The Replay client when playing audio **SHALL** include the UTC timestamp.

The Replay client when playing audio **SHALL** indicate the location where the audio was recorded (i.e. CWP1 G/G, CWP2 A/G, VCS G/G1, VCS A/G23, RX34).

3.32 RTSP server generation of CRD statistics

The RTSP server **SHOULD** be able to output all CRD data stored over a configured period of time (i.e. daily, weekly and monthly) such that a CRD client can analyze data and generate a summary report of the CRD received.

An example of a report may include:

1. List and count of all G/G communications, party numbers, timestamps, etc.
2. List and count of all DA calls
3. List and count of IA calls
4. List and count of OVR calls
5. List and count of PTT activations, PTTid, frequency, timestamps, etc.
6. List and count of SQU activations, frequency, timestamps, etc.

3.33 RTSP server archiving

The RTSP server **SHALL** provide a method such that all stored audio and CRD data can be archived. Archiving **SHOULD** automatically occur when the server has reached a pre-defined quota of its available storage capacity.

4.0 RECORDING SPECIFICATION

4.1 Scope of specification

The scope of this specification is to define the Audio, Signaling and Management interworking between a client at a Voice Communication System (VCS) or Radio endpoint and a client at a Recorder endpoint (RTSP server) connected through an IP network.

4.2 Applicable configurations

This specification defines the signaling interworking between clients at the VCS or Radio endpoints and the RTSP server in order to record audio and call record data for the following configurations:

- Ground-Ground communications between User Terminal clients (at the position) or at recording points at the VCS network interfaces.
- Air-ground communications between User Terminal clients (at the position) and Radio Transmitters/Receivers/Transceivers clients at the remote radio sites or at recording points at the VCS and Radio network interfaces.

4.3 Basic Protocol requirements

In order to initially establish a connection between VCS or Radio clients and an RTSP server and to transport both audio and signaling between these endpoints, the interfaces **SHALL** be capable of employing the following protocols in order to achieve the required functionality:

- the SIP protocol (RFC3261 [7]) **MAY** be employed to initially establish a SIP session between the VCS/Radio client endpoints and the RTSP Server (Recording equipment) endpoint;
- the RTSP protocol (RFC2326 [5]) **SHALL** be employed to initially establish an RTSP session between the VCS/Radio client endpoints and the RTSP (Recording) Server endpoint and subsequently used to Start and Stop audio recording, transfer Call Record Data relative to the communication in progress and finally to tear down the session if necessary.
- The audio transport **SHALL** be supported by the Real-time Transport Protocol (RTP) (RFC3550 [15]), and **MAY** be augmented by its associated Real Time Control Protocol (RTCP) to allow monitoring of the audio packets delivery.

In addition to the signaling functionality specified in this document, it is assumed that the IP AGVN components also include the following functionality:

- one or more physical interfaces on the IP network side supporting, through layer 1 and layer 2 protocols, IP as the network layer protocol and UDP (RFC 768 [1]) and TCP (RFC 793 [3]) as transport layer protocols. UDP being used for the transport of SIP signaling messages, TCP being used for the transport of RTSP messages and audio

The following provides an example of SIP session establishment from a User Terminal, VCS or Radio client to the Recorder server.

4.4 Audio Specifications

The voice **SHALL** be coded in accordance with ITU-T Rec. G.711 A-law (in Europe) or mu-law (in North America/Japan) coding.

4.5 Active Recording

Recording **SHALL** be based on active sessions opened from User Terminal clients to one recording device (or two devices required for redundancy).

Recording **SHOULD** be based on active sessions opened from VCS network interfaces, Radio Receiver/Transceivers or specific 3rd party devices to one recording device (or two devices required for redundancy).

Note 34.

Active means that any client that sends or receives media streams (i.e. audio) during an established communication is responsible for sending a copy of these audio streams to the recorder(s).

4.6 Dual RTSP sessions from position

Each position **SHALL** support up to two voice recording channels.

The specific selection of the audio sources and their assignment to the two voice recording channels will be ANSP-specific.

In case of recording G/G or A/G related communications at the Position, the User Terminal client **SHALL** send audio and Call Record Data (CRD) to the RTSP server (Recording equipment) over two independent RTSP sessions in order to achieve redundancy. Both audio streams **SHALL** be identical.

The RTSP server **SHALL** have the capability of recording the RTP audio stream either from RTSP session 1 or 2. In case of an RTSP session failure, the RTSP server **SHALL** continue to record from the second RTSP session.

In case of recording G/G or A/G related communications at the VCS/Radio client **SHALL** establish one RTSP session with a single RTSP server and **MAY** establish dual RTSP sessions.

4.7 Date & Timestamp Synchronization for audio

The used time source **SHALL** be synchronized to the ATSU time source. This is assumed to be Universal Time Coordinated (UTC) to the accuracy specified by ICAO Convention on International Civil Aviation, Annex 5: “Units of Measurement to be Used in Air and Ground Operations” [34].

4.8 RTSP addresses

Each RTSP server (Recorder equipment) **SHALL** have the capability of being assigned a unique RTSP address such that RTSP sessions can be established to it from the VCS client or from the Radio client. An example of an RTSP address is as follows: rtsp://ANSP-recorder2:554/iprecorder/ RTSP/1.0

4.8.1 RTSP address permissions list

Each User Terminal, VCS and Radio client **SHALL** have a RTSP address permissions list, defining the RTSP Server (Recorder equipment) to which it is allowed to establish RTSP sessions.

Each RTSP server (Recorder equipment) **SHALL** have a RTSP address permissions list, defining the User Terminal, VCS and Radio clients to which it is allowed to accept RTSP sessions requests.

Terminal/VCS/Radio clients and RTSP server (Recorder equipment) **SHALL** refuse RTSP session establishment attempts to/from RTSP addresses not defined in the permissions list.

It **SHALL** be possible to disable the RTSP address permissions list check at each VCS/Radio client and RTSP server. In this case the VCS/Radio clients and RTSP server (Recorder equipment) **SHALL** accept RTSP session establishment attempts to/from any RTSP address.

4.9 **OPTIONS message**

The OPTIONS message is mandatory for use by RTSP servers.

The OPTIONS message **SHALL** list all the messages supported by the RTSP server.

4.10 **Recording Model for Telephone communication**

User Terminals participating in a G/G communication session **SHALL** generate a single RTP audio stream containing the sum of the incoming (IN) and outgoing (OUT) audio streams at the position. It **SHALL** establish dual RTSP sessions with its nominated RTSP server per G/G communication for redundancy purposes. The summed RTP audio stream **SHALL** be transported over both RTSP sessions towards its nominated RTSP server.

VCS network interfaces participating in a G/G communication session **SHOULD** generate a single RTP audio stream containing the sum all incoming (IN) and outgoing (OUT) audio streams at the interface. It **SHOULD** establish a single RTSP session with its nominated RTSP server per G/G communication and **MAY** establish dual RTSP sessions with its nominated RTSP server for redundancy purposes.

Note 35.

In order to simplify the concept, the following diagrams show a single recorder equipment. All described procedures are identical to two or a defined number of recorders.

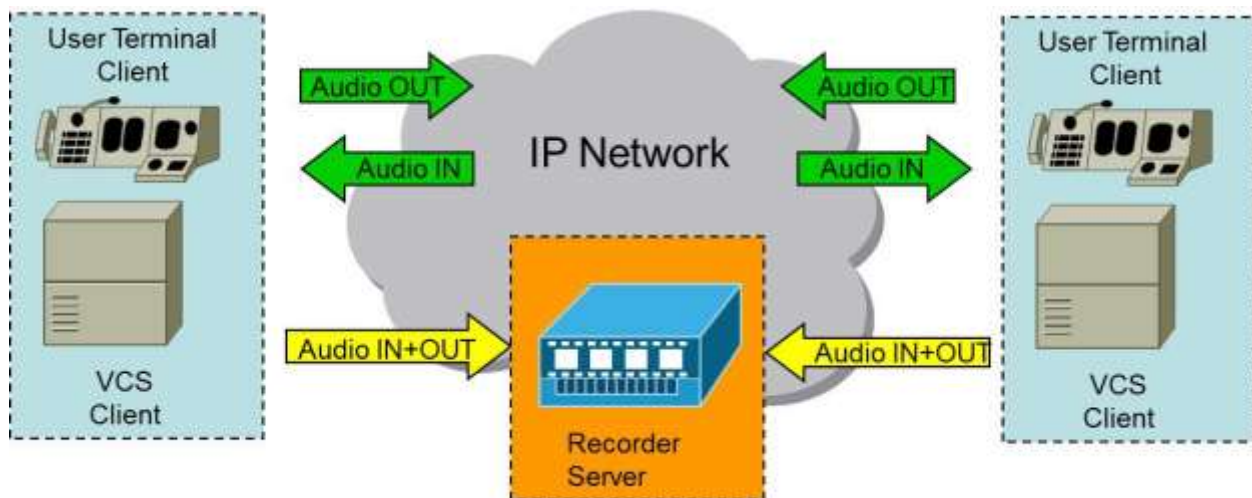


FIGURE 1: RECORDING TELEPHONE COMMUNICATION

4.11 Recording Model for Radio communication

User Terminals participating in an A/G communication session **SHALL** generate a single RTP audio stream containing the sum of all received (Rx) and transmitted (Tx) audio at the position from all configured frequencies. It **SHALL** establish dual RTSP sessions with its nominated RTSP server for redundancy purposes. The summed RTP audio stream **SHALL** be transported over both RTSP sessions towards its nominated RTSP server.

VCS network interfaces participating in an A/G communication session **SHOULD** generate a single RTP audio stream that contains either received (Rx) audio on the specific frequency and/or transmitted (Tx) audio on the specific frequency. It **SHOULD** establish a single RTSP session with its nominated RTSP server and **MAY** establish dual RTSP sessions with its nominated RTSP server for redundancy purposes.

Note 36.

It should be noted that recording audio on the Receive path only from a Radio Receiver will also include transmitted audio due to the radio receiver(s) picking-up the off-air transmitted audio from its associated Radio Transmitter. Transmitted audio recorded on the receive path is also proof that the audio was actually transmitted.

Radios (or Radio Gateways connecting legacy radios to an IP network) **MAY** provide a single audio stream that contains either the received (Rx) audio stream related to a single radio channel (applied to Radio Receivers only) or the summed transmit and receive (Rx+Tx) audio stream related to a single radio channel (applied to Radio Transceivers only).

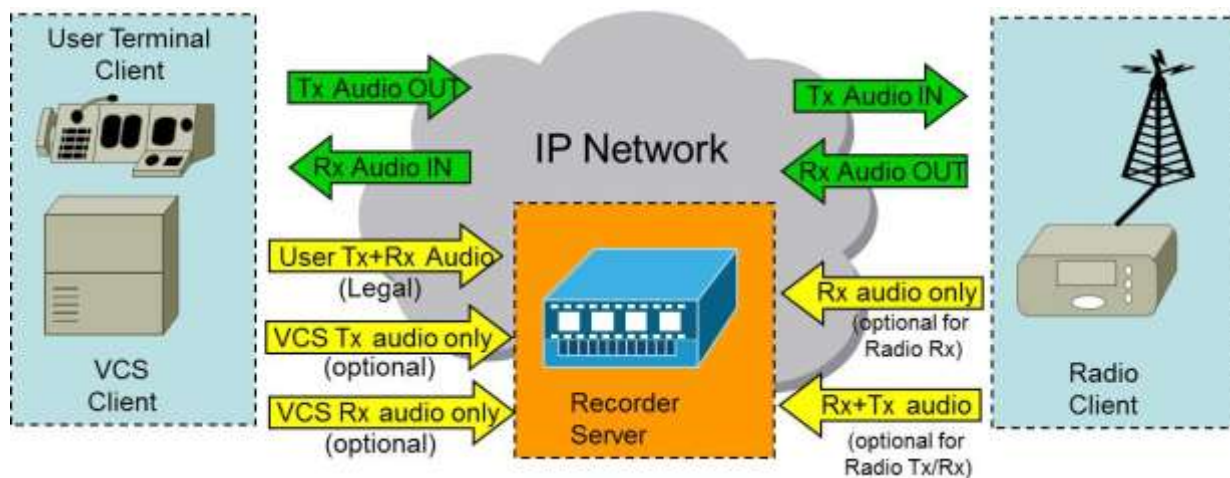


FIGURE 2: RECORDING RADIO COMMUNICATION

4.11.1 RTSP Header Extension

The RTSP server (recording equipment) implementation **SHALL** be ready to receive RTP packets with and without the RTP header extension being present.

The VCS/Radio client implementation **MAY** send RTP packets with or without the RTP header extension being present.

4.12 Session Initiation Protocol (SIP)

If SIP is employed (optional), the VCS/Radio and RTSP Server **SHALL** support SIP version 2 as specified in RFC 3261 [7].

The SIP protocol, is an application-layer control protocol which has been developed and designed within the IETF and is defined by RFC 3261 [7]. With respect to the RTSP server (Recording equipment) applications the SIP protocol **MAY** be used by a Voice Communication System (VCS) endpoint and/or Radio endpoint to establish, modify and terminate a SIP session with a RTSP server (Recording equipment) endpoint within an Air Traffic Services Ground Voice Network (AGVN).

Active recording requires an established session (i.e. a certain number of parameters that are exchanged between entities prior to any recording). User Terminal, VCS network interfaces, Radio and Recorder **SHALL** use RTSP for such sessions. As RTSP relies on a transport layer protocol (TCP or UDP), these entities **MAY** use SIP to exchange capabilities and connection information (i.e. IP address, port number, and transport protocol). In the case that SIP isn't used, it is necessary to define statically configured recorder binding [IP:Port] in order to employ RTSP implementations.

Once a communication session between VCS/Radio and RTSP endpoints has been established using the SIP protocol (optional), the two endpoints **SHALL** then establish an RTSP session (mandatory) and **SHALL** then employ the Real time Transport Protocol (RTP) (RFC 3550 [15]) for the transport of Audio in RTP packets between the endpoints (as defined in RFC 3550[15]) . The audio transport **MAY** be augmented by its associated control protocol (RTCP) (RFC 3550 [15]) to allow monitoring of voice packet delivery.

INVITE sip:recoder@atc.org SIP/2.0

...

Content-Type: application/sdp

Content-Length: 87

v=0

o=0 0 IN IP4 192.0.2.94

s=Recording

t=0 0

c=IN IP4 192.0.2.94

m=application 10554 rtsp rec

SIP/2.0 200 OK

...

Content-Type: application/sdp

Content-Length: 87

v=0

o=0 0 IN IP4 192.0.2.25

s=Recording

t=0 0

c=IN IP4 192.0.2.25

m=application 20554 rtsp rec

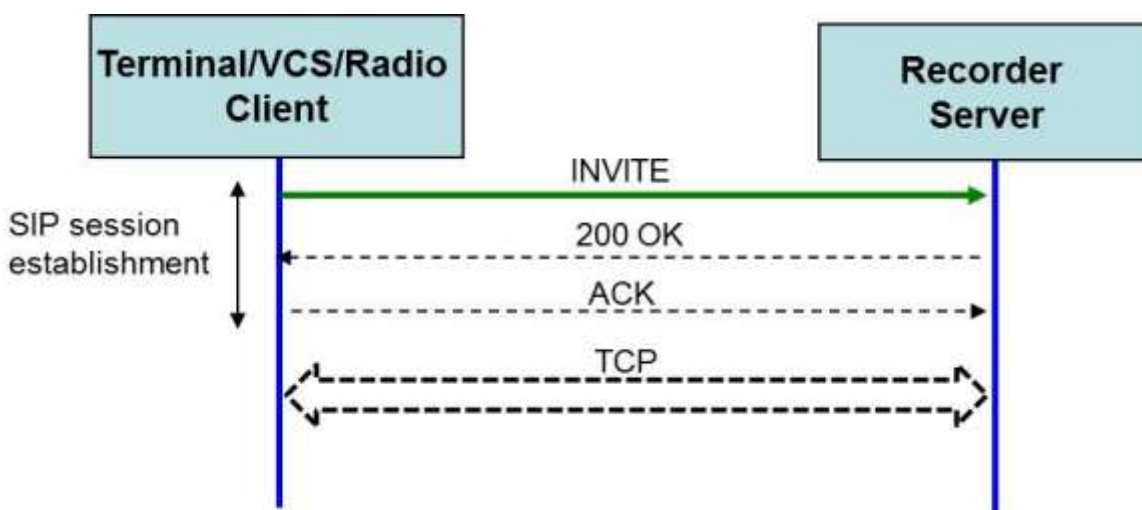


FIGURE 3: SIP SESSION ESTABLISHMENT: MESSAGE SEQUENCE

4.12.1 Logical SIP Entities

For the Recording Service within the IP AGVN, the logical SIP Entities like Registrar and Proxy Server are **OPTIONAL**. All User Agent services regarding Registration are also **OPTIONAL**.

Any entity involved in a recording session (User Terminal and Recorder) **SHOULD** register with a SIP Registrar using the REGISTER method according to RFC3261 [7]. It **SHOULD** be possible to register multiple contacts for a single Address of Record (AOR).

4.13 RTSP

User Terminals **SHALL** use RTSP to enable controlled, on-demand delivery of real-time data. Systems implementing RTSP **SHALL** support carrying RTSP over TCP and **MAY** support UDP. The default port for the RTSP server **SHALL** be 554 for both UDP and TCP.

The following assumes that the IP address of the Recorder is known and a TCP session has been established. Participants (User Terminals, Radios) **SHALL** use ANNOUNCE and SETUP to establish a recording session. Participants (User Terminals) **MAY** use DESCRIBE and SETUP to establish a replay session. This session setup provides the session description (connection information) and media description (media name and

transport address) of each participant. Participants (User Terminals, Radios) **SHALL** use TEARDOWN to close a session.

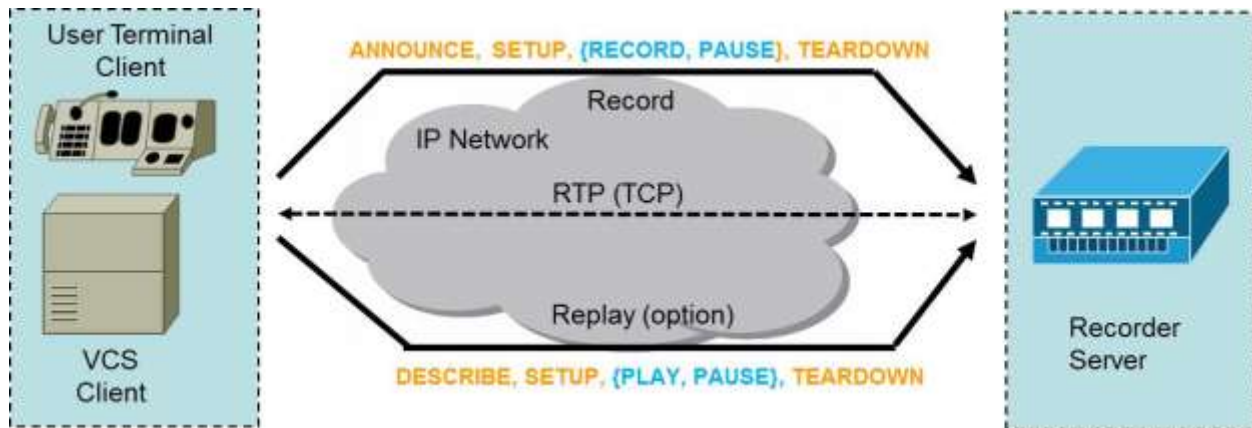


FIGURE 4: RTSP RECORD AND REPLAY SESSION

4.14 Transport

Transport of media **SHALL** be based on RTP over independent TCP as described later in this section. The Transport request and response header field indicates which transport protocol is to be used and configures its parameters such as destination address, compression, multicast time-to-live and destination port for a single stream. It sets those values not already determined by a presentation description.

Transports are comma separated, listed in order of preference. Parameters **MAY** be added to each transport, separated by a semicolon. The server **SHOULD** return a Transport response-header field in the response to indicate the values actually chosen. The Transport header field **MAY** also be used to change certain transport parameters. A server **MAY** refuse to change parameters of an existing stream.

The general syntax for the transport specifier is a list of slash separated tokens:

Value1/Value2/Value3...

Which for RTP transports take the form:

RTP/profile/lower-transport

The default value for the "lower-transport" parameters is specific to the profile. For RTP/AVP, the default is UDP for example.

RTP packets carrying relevant voice samples between point of voice presence and point of recording **MAY** be encrypted using strong encryption technology like 128Bit AES encoding. Stored Voice samples **SHALL** be encoded by 128/256 Bit AES algorithm.

The RTSP server (Recording equipment) **SHALL** have a mechanism to detect and report the loss of RTP packets in a stream received from VCS/GRS client.

4.14.1 RTP over independent TCP

This section adapts the guidelines for using RTP over TCP within SIP/SDP to work with RTSP as in [21].

There are two different methods for how to specify where the media should be delivered:

- **dest_addr:** The presence of this parameter and its values indicates the destination address or addresses (host address and port pairs for IP flows) necessary for the media transport.
- **No dest_addr:** The lack of the dest_addr parameter indicates that the server **SHALL** send media to the same address from which the RTSP messages originate. This does not work for transports requiring explicitly given destination ports.

A client codes the support of RTP over independent TCP by specifying an RTP/AVP/TCP transport option without an interleaved parameter. This transport option **SHALL** include the "unicast" parameter. If the client wishes to use RTP with RTCP, two ports (or two address/port pairs) are specified by the dest_addr parameter. If the client wishes to use RTP without RTCP, one port (or one address/port pair) is specified by the dest_addr parameter.

If the client wishes to play the active role in initiating the TCP connection, it **MAY** set the "setup" parameter on the Transport line to be "active", or it **MAY** omit the setup parameter, as "active" is the default. If the client signals the active role, the ports for all dest_addr values **SHALL** be set to 9 (the discard port).

If the client wishes to play the passive role in TCP connection initiation, it **SHALL** set the "setup" parameter on the Transport line to be "passive". If the client is able to assume the active or the passive role, it **SHALL** set the "setup" parameter on the Transport line to be "actpass". In either case, the dest_addr port value for RTP **SHALL** be set to the TCP port number on which the client is expecting to receive the RTP stream connection, and the dest_addr port value for RTCP **SHALL** be set to the TCP port number on which the client is expecting to receive the RTCP stream connection.

If upon receipt of a non-interleaved RTP/AVP/TCP request, a server decides to accept this requested option, the 2xx reply **SHALL** contain a Transport option that specifies RTP/AVP/TCP (without using the interleaved parameter, but using the unicast parameter).

The dest_addr parameter value **SHALL** be echoed from the parameter value in the client request unless the destination address (only port) was not provided, in which case the server **MAY** include the source address of the RTSP TCP connection with the port number unchanged.

In addition, the server reply **SHALL** set the setup parameter on the Transport line, to indicate the role the server will play in the connection setup. Permissible values are "active" (if a client set "setup" to "passive" or "actpass") and "passive" (if a client set "setup" to "active" or "actpass").

If a server sets "setup" to "passive", the "src_addr" in the reply **SHALL** indicate the ports the server is willing to receive an RTP connection and (in the case the client requested an RTCP connection by specifying two dest_addr ports or address/port pairs) an RTCP connection. If a server sets "setup" to "active", the ports specified in "src_addr" **SHALL** be set to 9.

The VCS and Radio clients **SHALL** have the "active" role by default, while the RTSP Server (i.e. Recorder) **SHALL** have the "passive" role. This implies that a Request from the VCS or Radio client **SHALL** always have the transport line defining the parameter "setup=active", while the Response from the Recorder **SHALL** always have the transport line defining the parameter "setup=passive".

The following illustrates a client server session example using RTP over independent TCP.

SETUP rtsp://recorder:554/iprecorder/ RTSP/1.0

CSeq: 1

Transport:

RTP/AVP/TCP;unicast;mode="RECORD";dest_addr=":9";setup=active;connection=new

RTSP/1.0 200 OK

CSeq: 1

Session: c408358f-a233-4dd2-9fb6-a338953cc8b2

Transport: RTP/AVP/TCP;unicast;dest_addr=":9";src_addr="192.0.2.5:9000";setup=passive
connection=new;ssrc=93CB001E

4.14.2 Framing Method

A 16-bit unsigned integer LENGTH field, coded in network byte order (big-endian), begins the frame. If the LENGTH is non-zero, an RTP or RTCP packet follows the LENGTH field. The value coded in the LENGTH field **SHALL** equal the number of octets in the RTP or RTCP packet. Zero is a valid value for LENGTH, and it codes the null packet, as in RFC4571 [25].

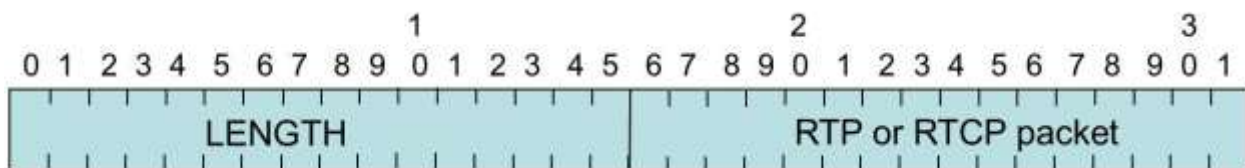


FIGURE 5: TCP FRAME FORMAT

Audio packets that are transported by an "RTP over independent TCP connection" **SHALL** be framed using the protocol defined in RFC4571 [25].

4.14.3 Lost audio prevention during RTSP session teardown

With RTSP signaling and RTP media being split on different sockets (as in the case of RTP over independent TCP), the RTSP server **SHALL NOT** immediately close the RTP socket when it receives a TEARDOWN message from a client. In order to ensure that the RTSP server receives all packets in the audio stream (including late arrivals due to re-transmissions), the RTSP server **SHALL** wait a pre-set time (i.e. 5 to 10 seconds), for the client to close its RTP socket first and stop sending RTP audio packets towards the RTSP server.

4.14.4 RTSP session timeout

The Keep-alives are assured on a TCP connection by configuring the values below from application when the TCP socket is created or directly from the operating system:

- TCP_KEEPIIDLE (ex. 10 seconds) maximum idle time (in seconds) before start sending keep-alive probes;
- TCP_KEEPINTVL (ex. 2 seconds) interval (in seconds) between keep-alive probes
- TCP_KEEPCNT (ex. 5 seconds) maximum keep-alive probes before assuming the connection is lost.

A half-open connection can occur when the Client or RTSP server closes or aborts the connection without the other side knowing about it. This could happen if, for example, one device had a software crash and was restarted in the middle of a connection, or if a glitch caused the states of the two devices to become unsynchronized.

When a client or RTSP server stops receiving Keepalives and the TCP_KEEPCNT expires, it **SHALL** start the process to abort the TCP connection. A TCP reset (TCP segment with RST flag=1) **SHALL** be used to terminate the TCP connection.

4.15 RTSP Control Messages

4.15.1 ANNOUNCE and SETUP

These messages **SHALL** be used to establish a recording session. The message body of ANNOUNCE **SHALL** contain a description of the media referenced by the requested URL, (e.g. rtsp://recorder:554/iprecorder/) using SDP, as in RFC3264 [8].

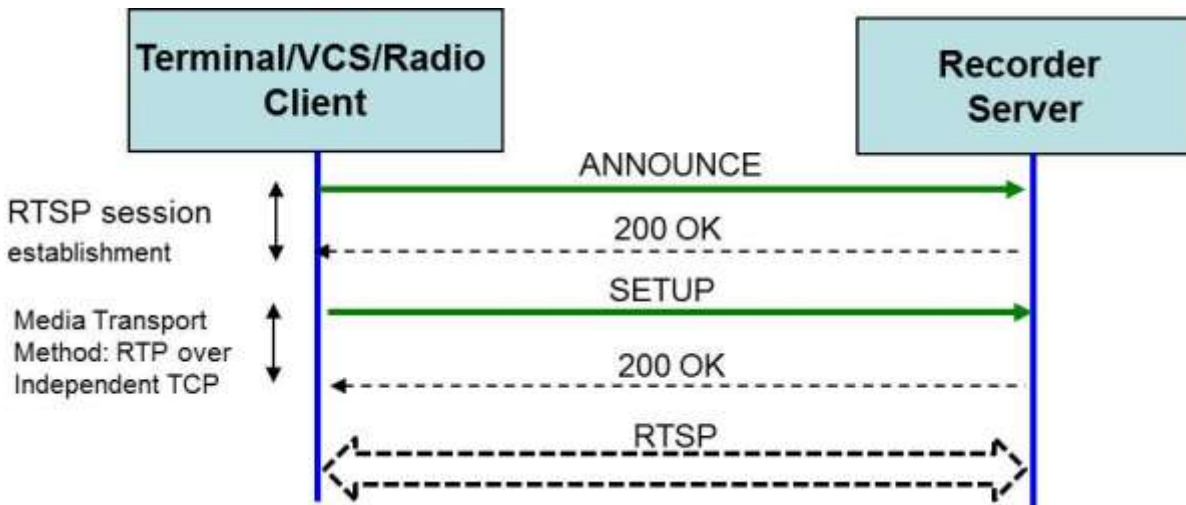


FIGURE 6: RTSP SESSION ESTABLISHMENT: MESSAGES AND SEQUENCE

The following gives an example of an RTSP session setup using “RTP over independent TCP” (i.e. non-interleaved binary data) request (in blue) and response (in red):

```

ANNOUNCE rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 1
Content-Type: application/sdp
...
v=0
o=first 2520644554 2838152170 IN IP4 first.example.net
s=Example
t=0 0
c=IN IP4 192.0.2.105
m=audio 0 RTP/AVP 0
a=rtpmap:0 PCMU/8000
...

```

```

RTSP/1.0 200 OK
CSeq: 1
...

```

```

SETUP rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 2
Transport: RTP/AVP/TCP;unicast;mode="RECORD";dest_addr=":9";setup=active;connection=new

```

```

RTSP/1.0 200 OK
CSeq: 2
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Transport: RTP/AVP/TCP;unicast;dest_addr=":9";src_addr="192.0.2.5:9000";setup=passive;
connection=new;ssrc=93CB001E

```

4.15.2 **RECORD**

This message **SHALL** be used to start data transmission on the stream allocated via SETUP. Clients (Terminals) **MAY** offer a connection reference (connref) to the recorder using an XML encoded message body (see section 0 and 4.20 for details). Clients **SHALL** always be expected to provide a connection reference in their initial request. Clients **MAY** submit “Call Record Data” (CRD) at this stage using the defined XML structure (see section 0 and 4.20 for details) within the RECORD message and **SHALL** always use the connref parameter.

The connref value provided by the client (request) **SHALL** be the same as that used by the server (recorder) in its response. The following gives an example about how to start recording including a client generated connection reference value (request in blue and response in red):

RECORD rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 2
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Content-Type: application/x-crd+xml

<call-record-data connref="403C232A-C510-45C7-973E-D55F5CF996AF" />
(see sections 0 and 4.20 for content details)

RTSP/1.0 200 OK
CSeq: 2
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Content-Type: application/x-crd+xml

<call-record-data connref="403C232A-C510-45C7-973E-D55F5CF996AF" />
(see sections 0 and 4.20 for content details)

The RTP stream **SHALL** be sent after a RECORD message is sent.

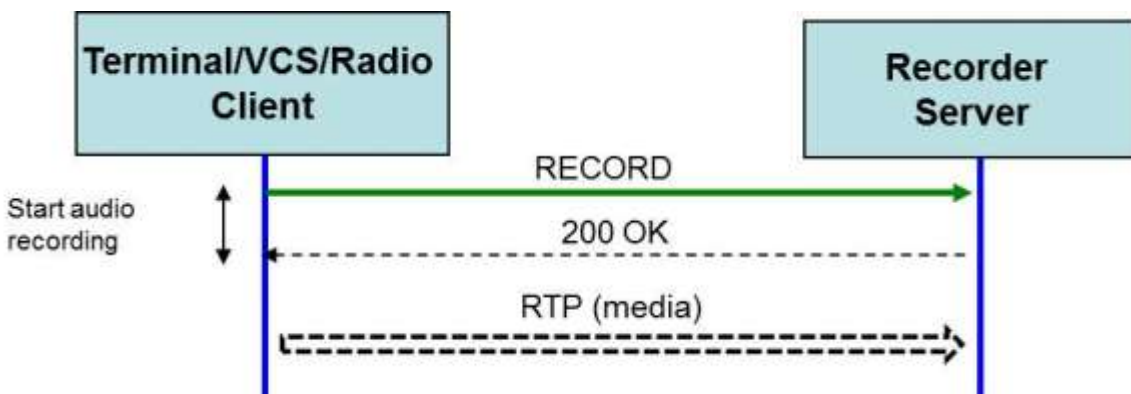


FIGURE 7: RECORD: MESSAGES AND SEQUENCE

4.15.3 PAUSE

This message **SHALL** be used to interrupt data transmission on the RTP stream that was previously started by the RECORD message. (request in blue and response in red):

PAUSE rtsp://recorder:554/iprecorder/ RTSP/1.0
 CSeq: 2
 Session: c408358f-a233-4dd2-9fb6-a338953cc8b2

RTSP/1.0 200 OK
 CSeq: 2
 Session: c408358f-a233-4dd2-9fb6-a338953cc8b2

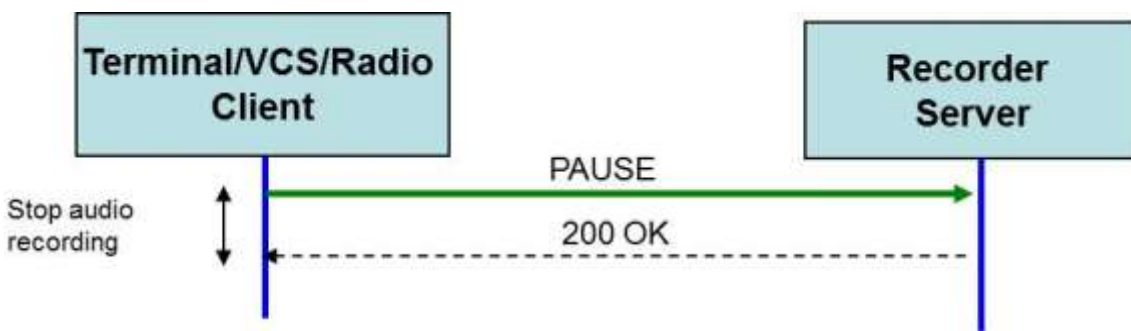


FIGURE 8: PAUSE: MESSAGES AND SEQUENCE

The RTP stream **SHALL** be interrupted after a PAUSE message is sent.

4.15.4 SET PARAMETER

This message **SHALL** be used to set the value of a parameter (Call Record Data) for a presentation or stream specified by the URI (request and response).

A SET_PARAMETER with a CRD body **SHALL** be allowed to be sent without a session identifier at any time, to request the recorder to store meta data to the specified connection as an atomic operation. Such an operation does not 'start' or 'stop' a connection, it simply adds information.

In this way it will be possible:

- to record data of calls that have not been successfully set up (ex. gateway is blocked, congestion, access list reject);
- to record data from the client without having to wait for a session setup to complete;
- after the call has been cleared and the corresponding RTSP session has been torn down, the operator may add some information to the call (ex. comments, ratings etc.);

Refer to Annex A.1.4 Recording a rejected call example for more information.

SET_PARAMETER

rtsp://recorder:554/iprecorder/ RTSP/1.0

CSeq: 3

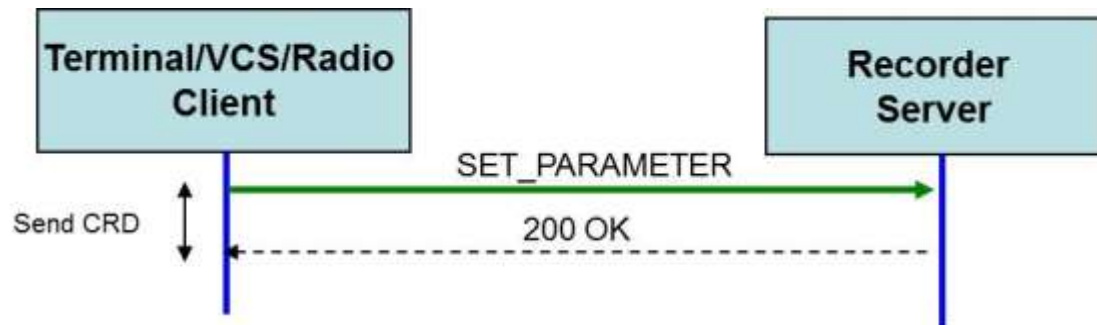
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2

Content-Type: application/x-crd+xml

*(see sections 0 and 4.20 for content details)***RTSP/1.0 200 OK**

CSeq: 3

Session: c408358f-a233-4dd2-9fb6-a338953cc8b2

**FIGURE 9: SET_PARAMETER: MESSAGES AND SEQUENCE****4.15.5 TEARDOWN**

This message **SHALL** be used to free resources associated with the stream specified by the URI (request in blue and response in red):

TEARDOWN rtsp://recorder:554/iprecorder/ RTSP/1.0

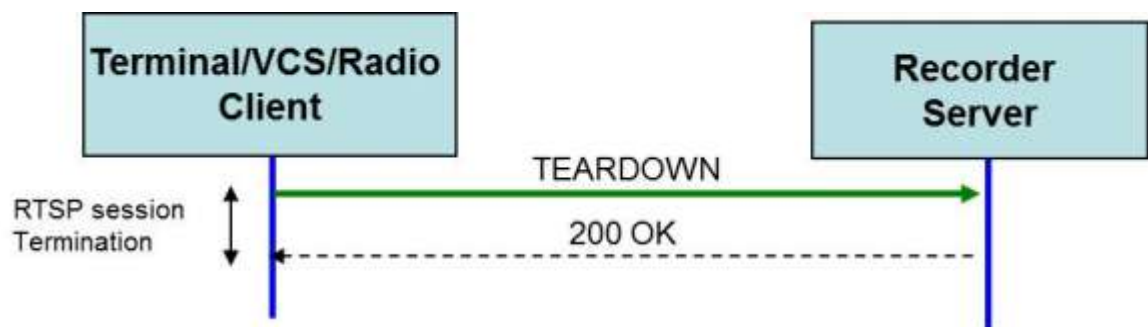
CSeq: 4

Session: c408358f-a233-4dd2-9fb6-a338953cc8b2

RTSP/1.0 200 OK

CSeq: 4

Session: c408358f-a233-4dd2-9fb6-a338953cc8b2

**FIGURE 10: TEARDOWN MESSAGES AND SEQUENCE**

4.15.6 **REPLAY**

This message sequence **SHOULD** be used to identify the audio message to be played, at a replay client, setup the ports and media transport method for the audio transfer, send the stored audio message (Play), pause the message (Pause) and finally teardown the session. It is comprised of DESCRIBE, SETUP, PLAY, PAUSE and TEARDOWN messages.

DESCRIBE rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 2
Accept: application/sdp

RTSP/1.0 200 OK
CSeq: 2
Server: Example Recorder
Content-Type: application/sdp
Content-Length: 157

v=0
o=unnamed 0 0 IN IP4 playback.example.net
s=Example Stream
t=0 0
a=range:npt=0.0-9.420000000
a=length:npt=9.420000000
m=audio 0 RTP/AVP 8
a=rtpmap:8 PCMA/8000

SETUP rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 3
Transport: RTP/AVP/TCP;unicast;mode="PLAY";dest_addr=":9";setup=active;connection=new

RTSP/1.0 200 OK
CSeq: 3
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
Server: Example Recorder
Transport: RTP/AVP/TCP;unicast;dest_addr=":9"/":9";src_addr="192.0.2.5:9000";setup=passive
connection=new;ssrc=93CB001E

PLAY rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 4
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
Range: npt=0-9.419000

RTSP/1.0 200 Success
CSeq: 4
Server: Example Recorder
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
Range: npt=0-9.419
RTP-Info:url=rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF;rtptime=3188274789;seq=4082

PAUSE rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 5

Session: 2da07059-961e-4998-81f8-0f6345e0b15f

RTSP/1.0 200 Success

CSeq: 5

Session: 2da07059-961e-4998-81f8-0f6345e0b15f

Server: Example Recorder

TEARDOWN rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0

CSeq: 6

Session: 2da07059-961e-4998-81f8-0f6345e0b15f

RTSP/1.0 200 OK

CSeq: 4

Session: c408358f-a233-4dd2-9fb6-a338953cc8b2

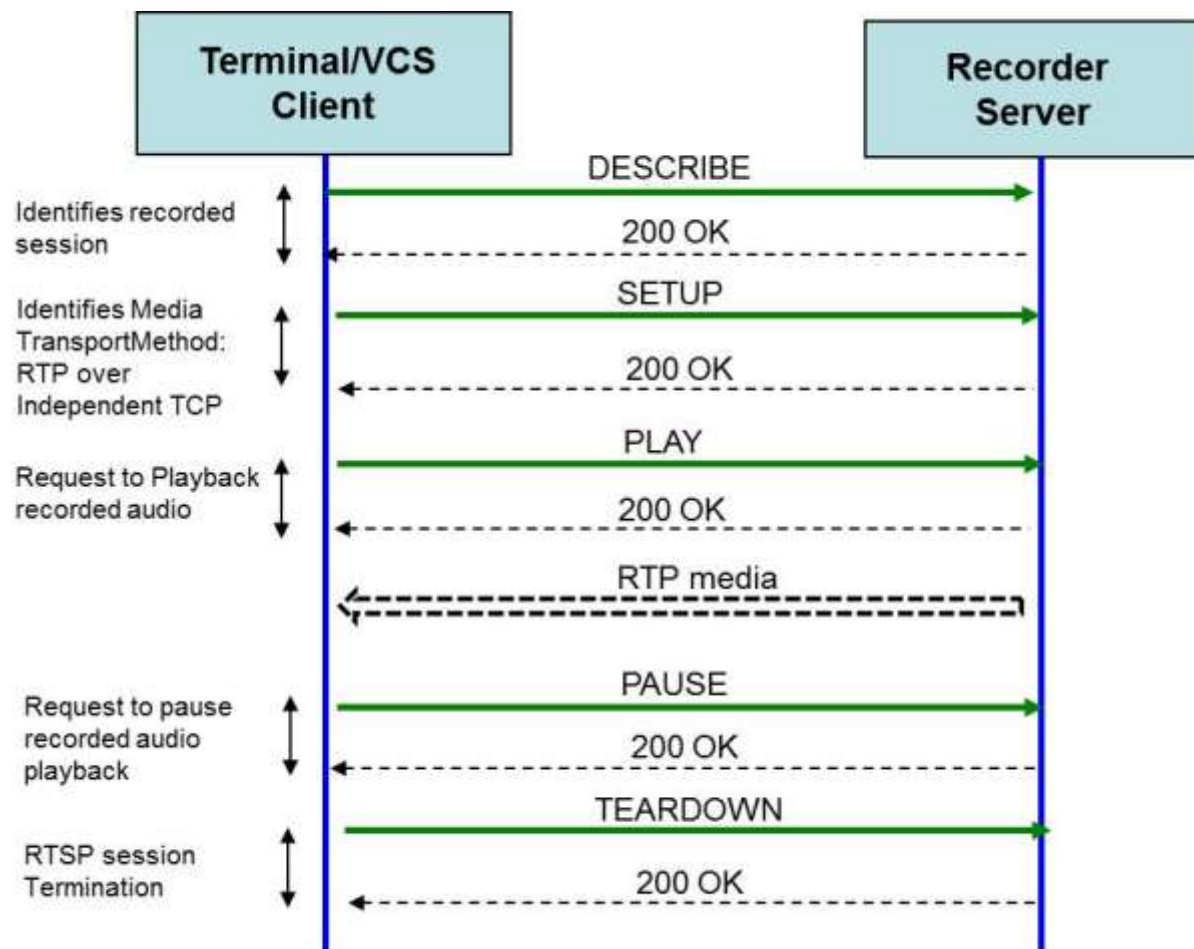


FIGURE 11: RTSP REPLAY SESSION: MESSAGES AND SEQUENCE

4.15.7 RTSP session signaling procedures between clients and RTSP server

1. The client is enabled to request establishment of an RTSP session to a RTSP server that has an RTSP address defined in its permissions list (if enabled);
2. The RTSP server is enabled to accept an establishment request for an RTSP session from a User Terminal, VCS or Radio client that has an RTSP address defined in its permissions list (if enabled);
3. For G/G communications a client is triggered to establish an RTSP session with its nominated RTSP server on outgoing G/G call initiation (i.e. sending SIP INVITE) or on incoming G/G call receipt (i.e. receiving a SIP INVITE);
4. For G/G communications one RTSP session **SHALL** therefore be established per incoming or outgoing telephone call by each User Terminal or VCS client to its nominated RTSP server.
5. Assuming that each User Terminal or VCS client will have its own nominated RTSP server (Recorder equipment), a G/G communication between clients would result in both having its own independent RTSP session with a RTSP server (recorder equipment).
6. For A/G communications a client is triggered to establish an RTSP session with its nominated RTSP server on SIP radio session initiation towards a radio (i.e. sending a SIP INVITE) or on receipt of a SIP radio session request from a VCS (i.e. receiving a SIP INVITE);
7. To initiate RTSP session establishment the client sends an RTSP ANNOUNCE message to the RTSP server.
8. An ANNOUNCE control message containing an SDP message body with a valid media description and rtpmap attribute appropriate for the audio codec G.711 PCM A-law or mu-law **SHOULD** therefore always be offered by the User Terminal, VCS or Radio client to the RTSP server.
9. When receiving an RTSP ANNOUNCE message from the User Terminal, VCS or Radio client, the RTSP server **SHOULD** verify that the RTSP address matches its own address. In the case that they are different the RTSP server can reject the message through a 404 “Not found” response.
10. When receiving an RTSP ANNOUNCE message, the RTSP server **SHOULD** also verify that it can accept the media description and rtpmap attribute in the SDP message body.
11. In the case that the media description and rtpmap attribute being offered by the User Terminal, VCS or Radio client is not supported by the RTSP server, it **SHOULD** reject the ANNOUNCE message with a “415 Unsupported Media type” response code by a “488 Not acceptable here” response or by a “606 Not Acceptable” response.
12. On receipt of a 4xx failure response to , the client **SHALL** indicate RTSP session failure.
13. In the case that the media description and rtpmap attribute being offered by the User Terminal, VCS or Radio client is supported by the RTSP server, a 200 OK success response to the ANNOUNCE message **SHOULD NOT** contain an SDP message body.

14. On receipt of 200 OK successful response the client sends an RTSP SETUP message to the RTSP server.
15. When receiving an RTSP SETUP message from the VCS or Radio client, the RTSP server **SHOULD** verify that the RTSP address matches its own address. In the case that they are different the RTSP server can reject the message through a 404 “Not found” response.
16. On receipt of a 4xx failure response, the client **SHALL** indicate RTSP session failure.
17. On receipt of 200 OK successful response the RTSP session has been successfully established.
18. The RTSP server (Recorder equipment) will generate a session identifier in response to a SETUP request. A session identifier is chosen randomly by the RTSP server and is at least eight octets long to make guessing it more difficult. The RTSP server will assign a Session identifier in its 200 OK response to the SETUP message from the VCS or Radio client.
19. On RTSP session timing out, the User Terminal, VCS or Radio client **SHOULD** attempt to re-establish the RTSP session towards the RTSP server.
20. Once an RTSP session has been opened between User Terminal, VCS or Radio client and RTSP Server (i.e. recorder equipment), the RTSP Server **SHOULD** keep the session permanently open until instructed to tear down the session by the User Terminal, VCS or Radio client. The RTSP server **SHOULD NOT** therefore close the session due to internal inactivity timeouts.
21. For G/G communications a client is triggered to teardown an established RTSP session with its nominated RTSP server when a G/G communication is cleared by either party (i.e. on sending or receiving a BYE request) or in the case of call intrusion or conference, when the last call is cleared.
22. For A/G communications a User Terminal client is triggered to teardown an established RTSP session with its nominated RTSP server when the last frequency is unassigned at the position. In most cases the RTSP session in the case of radio can be considered to be permanent due to a position always have at least one (emergency frequency) assigned.
23. For A/G communications a VCS client is triggered to teardown an established RTSP session with its nominated RTSP server when the SIP radio sessions to the radio frequency it is recording has been terminated. At the VCS client there could be one RTSP session established for each radio frequency.
24. For A/G communications a Radio client is triggered to teardown an established RTSP session with its nominated RTSP server when the SIP radio session to the VCS has been terminated.

4.16 Call Record Data Format

The following XML structure **SHALL** be used to transmit call record data within a SET_PARAMETER message:

```
<call-record-data connref="403C232A-C510-45C7-973E-D55F5CF996AF">
  <properties>
    <property name="">value</property>
  </properties>
  <operations>
    <operation name="" time="[utc-datetime]">value</operation>
  </operations>
</call-record-data>
```

Call Record data **SHALL** be composed of properties and operations. Any timestamp **SHOULD** be set by the client since it has the exact time reference for any local event. If a timestamp value is omitted, the server **SHALL** use the message arrival timestamp.

4.16.1 Properties

Properties are single values that will not change during the lifetime of a connection and usually do not require a time reference, except for properties that are representing timestamp information.

A client **MAY** send an updated value of a property that is already set if the new value is a more accurate one. In that case the recorder **MAY** overwrite the previous value if present. The recorder does not need to hold any previous values since properties are only those values that can have only one instance for a connection. For example, the direction of the connection never changes during its lifetime.

Examples:

- Direction (of the connection): 0 = unknown (default), 1 = incoming, 2 = outgoing
<property name="Direction">1</property>
- Priority: 1 – emergency, 2 – urgent, 3 – normal, 4 – non-urgent<property name="Priority">3</property>
- CallingNr, CalledNr, AlertingNr, ConnectedNr: preferred in "tel:" URI format
<property name="CallingNr">tel:+4311503</property>
- SetupTime, AlertTime, ConnectTime, DisconnectTime: utc-datetime
<property name="SetupTime">2007-08-01_05-50-00.789+0000</property>
- DisconnectCause: Cause values according to ITU-T Rec. Q.931
<property name="DisconnectCause">19</property>
- DisconnectSource: 0 = unknown (default), 1 = calling side, 2 = called side, 3 =other
<property name="DisconnectSource">1</property>
- Call Type: Classification of call type, 1 = DA call, 2 = IA call, 3 = OVR call
<property name="Type">1</property>
- CallRef: a common CRD identifier between calling and called party for one telephone

call

```
<property name="CallRef"> 583a74a2-0c8b3964-020acf0e-0a622c2f@192.168.1.193
</property>
```

- ThreadRef: a common Thread Reference identifier between calling and called party for one telephone call or SIP Radio session. Value remains the same during a call transfer.

```
<property name="ThreadRef"> 692b89d1-1b123981-134af0d1-9e533b1f@192.168.1.193
```
- ```
</property>
```
- PTTid: PTT=id value assigned by Radio to VCS client when establishing SIP session  

```
<property name="PTTid"> 000001 </property>
```

#### 4.16.2 Operations

Operations are events during the lifetime of a connection that may happen at any time and **SHOULD** be preserved at the recorder. Examples:

- RedirectedNr: Representing a "tel:" URI format to notify a redirection with the new target.  

```
<operation name="RedirectedNr"
time="20070801T054035Z456">+431156</operation>
```
- TransferredNr: Representing a "tel:" URI format to notify a change in the Contact address header with the conference URI.  

```
<operation name="TransferredNr"
time="20070801T054035Z456">+431156</operation>
```
- HOLD indicate when a call is put on hold- 0=call not on-hold, 1=call put on-hold by calling party, 2=call put on-hold by called party.  

```
<operation name="HOLD" time="2007-08-01_05-50-00.789+0000">2</operation>
```
- PTT: change of PTT state  

```
<operation name="PTT" time="2007-08-01_05-50-00.789+0000">0</operation>
```
- SQU: change of SQU state  

```
<operation name="SQU" time="2007-08-01_05-50-00.789+0000">0</operation>
```
- PTTS: change of PTTS state  

```
<operation name="PTTS" time="2007-08-01_05-50-00.789+0000">1</operation>
```
- PM: change of PM state  

```
<operation name="PM" time="2007-08-01_05-50-00.789+0000">1</operation>
```
- VOTING indicate selected Rx by voting algorithm  

```
<operation name="VOTING" time="2007-08-01_05-50-00.789+0000">1</operation>
```

#### **4.16.3 Time Stamp format**

The UTC timestamps defined by RECORD, PAUSE, SET\_PARAMETER and TEARDOWN CRD XML body messages **SHALL** take the following format:

- YYYY-MM-DD\_HH:MM:SS.XXX+0000
- YYYY = year, four digits
- MM = month, two digits
- DD = day, two digits
- HH = hours, two digits
- MM = minutes, two digits
- SS = seconds, two digits
- XXX = milliseconds, three digits
- +0000 MUST be always fixed since the time stamp sent to the REC is in UTC format no matter what local time is in client equipment

Example:

2011-05-16\_08:30:30.123+0000

#### **4.17 Referencing Call Scenarios**

##### **4.17.1 Connection Reference generation**

In order to have a unique connref, the IP address of the client **SHALL** be added when connref is generated.

##### **4.17.2 Call Reference generation (Telephone only)**

For telephone calls the SIP Call-ID (as defined by the SIP Callid header within the INVITE message) **SHALL** always be used for the CallRef value. The VCS client that initiates a call by sending an INVITE will therefore be responsible for generating the CallRef.

In case of transit (middle) equipment that has two SIP call legs, CallRef is generated by concatenating the SIP Call-ID from both legs.

##### ***Note 37.***

*For example endpoint A calls endpoint B via transit system T. Assuming in this case that recording is enabled at systems A, B and T. For A and B endpoints it is straightforward which SIP Call-ID is to be used as CallRef because this equipment has only one SIP call leg. In case of the transit system T however there are two SIP call legs established one with A and one with B. In this case the transit system T will concatenate SIP Call-IDs from both call legs when CallRef is sent to its RTSP server (recording equipment).*

Generation of Call-Ref for other telephony calls that are not established via SIP is outside of scope.

#### 4.17.3 Thread Reference generation

For both telephone calls and radio sessions a unique ThreadRef value **SHOULD** be included in an INVITE message. The VCS client that initiates a call or radio session by sending an INVITE will therefore be responsible for generating the ThreadRef. The format proposed for a ThreadRef is a format similar to that for the CallRef.

```
<property name="ThreadRef">XXXXXXXXX-22222222@192.168.1.10</property>
```

#### 4.17.4 ConfRef generation for conference

Currently only the RTSP server of the VCS client with the isfocus role, is able to determine that there is a conference in progress, by the fact there are multiple CRDs being received from the VCS client on the same RTSP connection over a pre-defined time. As multiple CRDs can also occur in the case of a Call Transfer, it is difficult to distinguish between Conference and Call Transfer for example.

When a VCS client with an isfocus role initiates a conference, it **SHOULD** include a ConfRef (as well as a CallRef and ThreadRef etc) in the INVITE request. The CallRef and ThreadRef generated will therefore be different for each call in the conference, but the ConfRef would be common to all calls in the conference. When a VCS client as isfocus receives a request to join the conference, it **SHOULD** include the ConfRef in a 200 OK.

All VCS clients involved in a Conference will therefore have a common reference for the conference. Each VCS client **SHOULD** then include the same ConfRef in the CRD to its RTSP server. The RTSP server associated with each VCS client in the conference can then determine that received audio is a summed stream from the conference.

#### 4.17.5 CRD XML body in RECORD, PAUSE, SET PARAMETER and TEARDOWN

CRD relating to telephone calls **SHOULD** be sent at call initiation, call accept/refusal and at call termination.

The call initiation CRD **SHOULD** be included in:

- SET\_PARAMETER message with XML body that follows the initial SETUP message without XML body

The call accept CRD **SHOULD** be included in either:

- RECORD with XML body
- SET\_PARAMETER with XML body, RECORD **SHALL** still be sent when call is accepted

The call termination CRD **SHOULD** be included in either:

- PAUSE with XML body followed by a TEARDOWN message without XML

- body
- SET\_PARAMETER with XML body, PAUSE and/or TEARDOWN **SHALL** still be sent when call is terminated
- TEARDOWN with XML body without a previous PAUSE message being sent

#### 4.17.6 SETUP without Call Record Data

As the SETUP message may be parsed by external device (ex. firewalls), the implementation of this message **SHALL** be strictly compliant with RFC 2326 [5] to avoid any interoperability issue. The SETUP message **SHALL** therefore carry only transport information and **SHALL NOT** include an XML message body containing CRD.

#### 4.18 WG67-Version

The RTSP WG67-Version header field **SHALL** appear in any request and in any response, but RTSP elements need to be prepared to receive messages without that header field indicating implementations based on earlier WG-67 EDs.

The syntax of the RTSP header field **SHALL** have the following content:

```
WG67-Version = "WG67-Version" HCOLON version-value *(COMMA
version-value)
version-value = field-value *(SEMI version-params)
field-value = type "." number
type = "radio" / "phone" / "legacy-eu" / "recorder" /
"supervision"
number = 2*DIGIT
```

The field-value **SHALL** contain the latest document version that reflects the implemented version of the corresponding EUROCAE ED as listed in para. 2.2 **Error! Reference source not found..** Implementations **SHALL** be able to process multiple header field rows with the same name in any combination of the single-value-per-line or comma-separated value forms. According actions **MAY** be specified where applicable.

Examples are:

```
WG67-Version: recorder.01
```

```
WG67-Version: recorder.01; gateway="RG"
```

#### **Note 38.**

*Version-params may be used for future extensions and are not described further in this document.*

## 4.19 Telephone Call Record Data

### 4.19.1 Telephone Call Record Data Properties

User Terminals (T) **SHALL** and VCS network interfaces (V) **SHOULD** transmit the following properties to the Recorder using SET\_PARAMETER.

**TABLE 1- LIST OF TELEPHONE PROPERTIES**

Property	Format	Description/Example	Source	Requirement
Direction	INTEGER	0...unknown, 1...incoming, 2...outgoing	T or V	mandatory
Priority	INTEGER	1 - emergency 2 - urgent 3 - normal 4 - non-urgent	T or V	mandatory
CallingNr	TEL URI	<a href="tel:+4311503">tel:+4311503</a>	T or V	mandatory
CalledNr	TEL URI	tel:+4311503	T or V	mandatory
AlertingNr	TEL URI	<a href="tel:+4311503">tel:+4311503</a>	T or V	optional
ConnectedNr	TEL URI	tel:+4311503	T or V	optional
SetupTime	UTC DATETIME	2007-08- 01_05:40:30.123+0000	T or V	mandatory
AlertTime	UTC DATETIME	2007-08- 01_05:40:30.123+0000	T or V	optional
ConnectTime	UTC DATETIME	2007-08- 01_05:40:30.123+0000	T or V	optional
DisconnectTime	UTC DATETIME	2007-08- 01_05:40:30.123+0000	T or V	mandatory
DisconnectCause	INTEGER	As defined by ITU-T Rec. Q.931	T or V	optional
DisconnectSource	INTEGER	0...Unknown 1...calling side, 2...called side 3...other	T or V	mandatory
Call Type	INTEGER	1...DA call 2...IA call 3...OVR call	T or V	optional
CallRef	STRING	583a74a2-0c8b3964-020acf0e- 0a622c2f@192.168.1.193	T or V	optional
ThreadRef	STRING	692b89d1-1b123981-134af0d1- 9e533b1f@192.168.1.193	T or V	optional
ConfRef	STRING	784cd851-6f65416-543af0e1- 7b8761fb@192.168.1.193	T or V	optional

**TABLE 2- CALL TIMER PROPERTIES AND MEANING**

Timer property	Meaning
SetupTime	Calling client: time when a SIP INVITE request is sent. Called client: time when a SIP INVITE request is received.
AlertTime	Calling client: time when a 180Ringing response is received. Called client: time when a 180Ringing response is sent.
ConnectTime	Calling client: time when a 200 OK response is received. Called client: time when an ACK to the previously sent 200 OK response is received
DisconnectTime	Clearing client : time when a SIP BYE request is sent for an established point-to-point communication. Cleared client: time when a SIP BYE request is received for an established point-to-point communication. Clearing conference: time when the last SIP BYE request is sent in a conference (when last two users remain) Call Reject: time of the 4xx/5xx/6xx response. .

**4.19.2 Telephone Call Record Data Operations**

User Terminals (T) **SHALL** transmit the following operations to the Recorder using SET\_PARAMETER. Note: Operations include per definition a UTC date-time reference as unique timestamp.

**TABLE 3- LIST OF TELEPHONE OPERATIONS**

Operation	Format	Description/Example	Source	Requirement
RedirectedNr	TEL URI	tel:+4311503	T	mandatory
TransferredNr	TEL URI	tel:+3411503	T	optional
HOLD	INTEGER	0 - hold off, call is not on hold 1 - call has been put on hold by calling party 2 - call has been put on hold by called party	T	optional

**Note 39.**

*The TransferredNr operation may be used to indicate when a SIP contact header is changed via following a re-INVITE (e.g. used during a conference). A normal call can be added to a conference hosted at the same position by using a re-INVITE, however its SIP contact address will change to indicate that of the conference address.*

#### **4.19.3 CRD signaling procedures between clients and RTSP server for G/G communications**

1. When a client initiates or receives a G/G communication (i.e. sends a SIP INVITE or receives a SIP INVITE) it **SHOULD** also establish RTSP session(s) to its RTSP server (i.e. Recorder). To establish RTSP session(s), it sends a RTSP SETUP message without XML body followed by a SET\_PARAMETER message with XML body (containing Call Record Data) to the RTSP server. Each SETUP or SET\_PARAMETER message will be acknowledged by a 200 OK response.
2. When a G/G communication is manually or automatically answered by the called party there will be a 200 OK response received or sent to the initial SIP INVITE and this **SHOULD** trigger the client to send either a RECORD message with XML body to its RTSP server or a SET\_PARAMETER message with XML body followed by a RECORD message without XML body. The client **SHOULD** then sum any incoming and outgoing audio and send this as one stream towards its RTSP Server(s).
3. Audio packets (summing audio from parties) will start to be sent to the RTSP server(s) when a RECORD message is sent and not when a 200 OK acknowledgement is received from the RTSP server.
4. When a client sends or receives a BYE message to terminate an established G/G communication, this **SHOULD** also trigger the client to terminate the RTSP session(s) to the RTSP server. On clearing the G/G communication, either:
  - a. the PAUSE message with XML body will be sent prior to a TEARDOWN message without XML body or
  - b. the SET\_PARAMETER with XML body will be sent prior to a PAUSE and/or a TEARDOWN message without XML body or
  - c. the TEARDOWN with XML body without a previous PAUSE message being sent.

The sending of a PAUSE or TEARDOWN message will stop the sending of audio packets to the RTSP server(s).

5. Audio packets (summing audio from parties) will stop to be sent to the RTSP server when a PAUSE or TEARDOWN message is sent and not when a 200 OK is received from the RTSP server(s).

**4.19.4 CRD signaling procedures between clients and RTSP server for a conference**

1. In the case of a conference, initiating the 1<sup>st</sup> call will cause the VCS client to establish an RTSP session(s) to the RTSP server. A SETUP message without XML body followed by a SET\_PARAMETER message with CRD XML body is sent to the RTSP server to establish an RTSP session(s).
2. When the 1st call is accepted, either a RECORD message with XML body or a SET\_PARAMETER with XML body followed by a RECORD message without XML body, sent to the RTSP server will imply summed audio is being sent to the RTSP server.
3. When the 2nd call of the conference call is initiated it will trigger a SET\_PARAMETER with XML body to be sent to the RTSP server. When the 2<sup>nd</sup> call is accepted and brought into the conference, it will trigger another SET\_PARAMETER with XML body to be sent to the RTSP server. The audio from all connected parties will be summed and sent to the RTSP server.
4. The same methodology as previous point can be used to connect further parties to the conference call.
5. When a party leaves the conference it will trigger a SET\_PARAMETER message with XML body to be sent to the RTSP server. This will inform the RTSP server that a party has left the conference. The audio from this party will therefore no longer be mixed with that from other parties and sent to the RTSP server.
6. When the last remote party leaves the conference, this will trigger a TEARDOWN message with CRD XML body to be sent to the RTSP server. This will inform the RTSP server that the conference is terminated.

#### **4.20 Radio Call Record Data**

Call Record Data (CRD) relating to A/G communications **SHALL** be sent to the RTSP server:

- at SIP Radio session establishment/teardown
- each time there is a change in control bits being sent to the Radio or Radio Remote Control Equipment (RRCE) and when there is a change in the status indication bits being received from the Radio or RRCE.

The SIP radio session initiation CRD **SHOULD** be included in the initial SETUP message with XML body.

PTT or SQU activation CRD **SHOULD** be included in:

- a RECORD message with XML body

PTT or SQU deactivation CRD **SHOULD** be included in:

- a PAUSE message with XML body

Change in control bits at any time (i.e. PTTS, PM, SCT or RRC Control bits etc)

- a SET\_PARAMETER message with XML body

The SIP radio session termination CRD **SHOULD** be included in either:

- PAUSE with XML body followed by a TEARDOWN message without XML body
- SET\_PARAMETER with XML body, PAUSE and/or TEARDOWN **SHALL** still be sent when call is terminated
- TEARDOWN with XML body without a previous PAUSE message being sent

#### 4.20.1 Radio Call Record Data Properties

User Terminals (T) **SHALL** and VCS Network interfaces (V), Radios (R) and Radio Remote Control Equipment (RRC) **SHOULD** transmit the following properties to the RTSP server (Recorder) using SET\_PARAMETER.

**TABLE 4- LIST OF RADIO PROPERTIES**

Property	Format	Description/Example	Source	Requirement
FrequencyID	STRING	118.005	T,V,R,RRC	mandatory
PTTid	INTEGER	000000....111111	T,V,R,RRC	mandatory
BSS Method	INTEGER	0...7	R,RRC	optional
CallingNr	TEL URI	tel:+4311503	T,V,R,RRC	optional
CalledNr	TEL URI	tel:+4311502	T,V,R,RRC	optional
Direction	INTEGER	0...unknown 1...incoming 2...outgoing	T,V,R,RRC	optional

#### 4.20.2 Radio Call Record Data Operations

User Terminals (T) **SHALL** and VCS network interfaces (V), Radio (R) and Radio Remote Control Equipment (RRC) **SHOULD** transmit the following operations to the Recorder using SET\_PARAMETER. Note: Operations include per definition a UTC date-time reference as unique timestamp.

**TABLE 5- LIST OF RADIO OPERATIONS (1)**

Operation	Format	Description/Example	Source	Requirement
PTT	INTEGER	0 - PTT OFF 1 - Normal PTT ON 2 - Coupling PTT ON 3 - Priority PTT ON 4 - Emergency PTT ON	T,V	mandatory
SQU	INTEGER	0 - OFF 1 - ON	R, RRC	mandatory for Radio and RRCE in single frequency mode
PTTS	INTEGER	0 - OFF 1 - ON	T,V,R,RRC	mandatory
PM	INTEGER	0 - OFF 1 - ON	T,V,R	mandatory
Simultaneous Transmission	INTEGER	0 - None detected 1 - Simultaneous transmission detected	R	mandatory
BSS Quality Index	INTEGER	-100...-70 (RSSI)	R,RRC	optional (also used in RRC single frequency mode)
VOTING	INTEGER	0 - voting disabled, default option 1 - voting enabled, current SQU connref selected by voting algorithm 2 - voting enabled, current	T	optional

Operation	Format	Description/Example	Source	Requirement
		SQU connref not selected by voting algorithm		

In addition the following operations relating to Radio Remote Control Equipment (RRCE) configured in Paired frequency mode only **SHOULD** be included:

**TABLE 6- LIST OF RADIO OPERATIONS (2)**

Operation	Format	Description/Example	Source	Requirement
SQF1	INTEGER	0 - OFF 1 - ON	RRCE	mandatory (RRCE paired frequency mode)
SQF2	INTEGER	0 - OFF 1 - ON	RRCE	mandatory (RRCE paired frequency mode)
BSS Quality Index SQI F1	INTEGER	-100...-70 (RSSI)	RRCE	optional (RRC paired frequency mode only)
BSS Quality Index SQU F2	INTEGER	-100...-70 (RSSI)	RRCE	optional (RRC paired frequency mode only)

In the case Radio Remote Control equipment (configured in either single or paired frequency mode) the following operations **SHALL** also be included within a SET\_PARAMETER message sent to the Recorder each time a RRC command message is sent or a RRC Response message is received.

- A copy of 8 RRC control bits within a RRC Command message sent to Radio Remote Control Equipment (RRCE) in the RTP Tx Header Extension Type 3, expressed as 8 Binary bits.
- A copy of the 8 RRC control bits within a RRC Response message received from the Radio Remote Control Equipment (RRCE) in the RTP Rx Header Extension Type 3, expressed as 8 Binary bits.

**TABLE 7- LIST OF RADIO OPERATIONS (3)**

Operation	Format	Description/Example	Source	Requirement
RRCC	INTEGER	Bit 12345678 00000000 - 11111111	RRCE	mandatory (RRCE in single and paired frequency mode)
RRCR	INTEGER	Bit 12345678 00000000 - 11111111	RRCE	mandatory (RRCE in single and paired frequency mode)

The eight RRC control bits from bit 1 to bit 8 have their meaning defined by the following table.

**TABLE 8- RRC CONTROL BITS**

<b>Bit</b>	<b>Definition</b>	<b>Value meaning</b>
1	MSTxF1 - Main/Standby Transmitter F1	MSTxF1=0 Main Tx F1 selected MSTxF1=1 Standby Tx F1 selected
2	MSRxF1 - Main/Standby Receiver F1	MSRxF1=0 Main Rx F1 selected MSRxF1=1 Standby Rx F1 selected
3	MSTxF2 - Main/Standby Transmitter F2	MSTxF2=0 Main Tx F2 selected MSTxF2=1 Standby Rx F2 selected
4	MSRxF2 - Main/Standby Receiver F2	MSRxF2=0 Main Rx F2 selected MSRxF2=1 Standby Rx F2 selected
5	SelTxF1 – Select Transmitter F1	SelTxF1=0 Tx F1 not keyed SelTxF1=1 Tx F1 keyed
6	SelTxF2 – Select Transmitter F2	SelTxF2=0 Tx F2 not keyed SelTxF2=1 Tx F2 keyed
7	MuRxF1 – Mute Receiver F1	MuRxF1=0 Rx F1 unmuted MuRxF1=1 Rx F1 muted
8	MuRxF2- Mute Receiver F2	MuRxF2=0 Rx F2 unmuted MuRxF2=1 Rx F2 muted

An RRCE configured for selection of all main systems, with keying enabled at both transmitters and both receivers unmuted, would have the 8 bits set to the following:  
00001100

An RRCE configured for selection of all standby systems, with keying enabled at both transmitters and both receivers unmuted, would have the 8 bits set to the following:  
11111100

An RRCE configured for selection of all standby systems, with keying disabled F2 and receiver F2 muted, would have the 8 bits set to the following:  
11111001

#### **4.20.3 CRD signaling procedures between clients and RTSP server for A/G communications**

1. When a VCS sends a PTT activation or a Radio receives a PTT activation, it **SHOULD** be triggered to send a RECORD message with XML body to its RTSP server. A VCS or Radio **SHOULD** then send outgoing audio as one stream towards its RTSP Server. Each RECORD message would be acknowledged with a 200 OK response from the RTSP server.
2. When a VCS sends a PTT deactivation or a Radio receives a PTT deactivation, it **SHOULD** be triggered to send a PAUSE message with XML body to its RTSP server. A VCS or Radio **SHOULD** then stop sending audio towards its RTSP Server. Each PAUSE message would be acknowledged with a 200 OK response from the RTSP server.
3. When a VCS receives an SQU indication or Radio sends a SQU indication, it **SHOULD** trigger the VCS or Radio to send a RECORD message with XML body to its RTSP server. The radio **SHOULD** then send incoming audio as one stream towards its RTSP Server. Each RECORD message would be acknowledged with a 200 OK response from the RTSP server.
4. When a VCS stops receiving a SQU indication or Radio stops sending a SQU indication, it **SHOULD** trigger the Radio to send a PAUSE message with XML body to its RTSP server. A Radio **SHOULD** then stop sending audio towards its RTSP Server. Each PAUSE message would be acknowledged with a 200 OK response from the RTSP server.
5. Between a PTT deactivation event and a PTT activation event or between a SQU deactivation event and a SQU activation event there is no media being sent to the RTSP server as this is just silence or off-air noise. The RTSP server **SHOULD** save each radio message, but it isn't considered necessary to also insert silence periods between these Radio messages.
6. When a VCS or Radio detects a change in the PTTS, PM, SCT or RRC control bits, this **SHOULD** trigger it to send a SET\_PARAMETER message with XML body to its RTSP server. Each SET\_PARAMETER message would be acknowledged with a 200 OK response from the RTSP server.

## ANNEX A: Call Record Data message examples

### A.1. CRD messages for Basic Telephone Calls

#### A.1.1. Basic outgoing call – unanswered example

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when an outgoing call is initiated by sending a SIP INVITE. The RTSP server will respond to the RTSP SETUP request with a 200 OK response.

The following messages are then sent to the RTSP server.

**SET\_PARAMETER** (sent after 200 OK response received to RTSP SETUP)

```
<call-record-data connref="11111111-XXXXXXXX@192.168.1.10">
<properties>
<property name="CallRef">XXXXXXXX-11111111@192.168.1.10</property>
<property name="Direction">2</property>
<property name="Priority">3</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:30.123+0000</property>
</properties>
```

**TEARDOWN** (sent when call cleared on sending SIP CANCEL)

```
<call-record-data connref="11111111-XXXXXXXX@192.168.1.10">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:40.123+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>
</properties>
```

#### A.1.2. Basic incoming call – answered example

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when an incoming call indicated by SIP INVITE has been received. The RTSP server responds to the RTSP SETUP request with a 200 OK response.

The following messages are then sent to the RTSP server.

**SET\_PARAMETER** (sent after 200 OK response received to RTSP SETUP)

```
<call-record-data connref="11111111-XXXXXXXX@192.168.1.10">
<properties>
<property name="CallRef">XXXXXXXX-11111111@192.168.1.10</property>
<property name="Direction">1</property>
<property name="Priority">1</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:30.123+0000</property>
</properties>
```

**RECORD** (sent when call answered by sending 200 OK response to SIP INVITE)

```
<call-record-data connref="11111111-XXXXXXXX@192.168.1.10">
```

```

<properties>
<property name="ConnectedNr">tel:2222</property>
<property name="ConnectTime">2011-05-16_08:30:36.123+0000</property>
</properties>

```

**TEARDOWN** (sent when call cleared on sending or receiving SIP BYE)

```

<call-record-data connref="11111111-XXXXXXXX@192.168.1.10">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:50.123+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>

```

### A.1.3. Basic outgoing call – answered example

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when an outgoing call initiated by SIP INVITE has been sent. The RTSP server responds to the RTSP SETUP request with a 200 OK response.

The following messages are then sent to the RTSP server.

**SET\_PARAMETER** (sent after 200 OK response received to RTSP SETUP)

```

<call-record-data connref="11111111-XXXXXXXX@192.168.1.10">
<properties>
<property name="CallRef">XXXXXXXX-11111111@192.168.1.10</property>
<property name="Direction">2</property>
<property name="Priority">1</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:30.123+0000</property>
</properties>

```

**RECORD** (sent when call answered on receiving 200 OK response to sent SIP INVITE)

```

<call-record-data connref="11111111-XXXXXXXX@192.168.1.10">
<properties>
<property name="ConnectedNr">tel:2222</property>
<property name="ConnectTime">2011-05-16_08:30:36.123+0000</property>
</properties>

```

**TEARDOWN** (sent when call cleared on sending or receiving SIP BYE)

```

<call-record-data connref="11111111-XXXXXXXX@192.168.1.10">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:50.123+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>

```

**A.1.4. Recording a rejected call example**

There is no need to send SETUP and TEARDOWN messages to the RTSP server in this case since the call is rejected for different reasons before being sent to a user. Sending just the SET\_PARAMETER message is enough in this case.

**SET\_PARAMETER**

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="CallRef">XXXXXXX-1111111@192.168.1.10</property>
<property name="Direction">1</property>
<property name="Priority">3</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:30.123+0000</property>
<property name="DisconnectTime">2011-05-16_08:30:30.123+0000</property>
<property name="DisconnectCause">34</property>
<property name="DisconnectSource">1</property>
</properties>
```

**A.1.5. Unattended Call Transfer example**

In this example A calls B and B transfers A to C. Two CRDs will be sent from A to its RTSP server on the same RTSP session.

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server after 'A' sends the SIP INVITE request to 'B'. The RTSP server responds to the RTSP SETUP request with a 200 OK response.

The following messages are then sent to the RTSP server.

**SET\_PARAMETER** (sent after 200 OK response received to RTSP SETUP)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="CallRef">XXXXXXX-1111111@192.168.1.10</property>
<property name="Direction">2</property>
<property name="Priority">1</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:30.123+0000</property>
</properties>
```

**RECORD** (sent when call answered on receiving 200 OK response to sent SIP INVITE)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="ConnectedNr">tel:2222</property>
<property name="ConnectTime">2011-05-16_08:30:36.123+0000</property>
</properties>
```

When A receives SIP REFER from B, A will make a new call to C.

**SET\_PARAMETER** (sent when INVITE is sent to C on same RTSP session)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="CallRef">XXXXXXX-22222222@192.168.1.20</property>
<property name="Direction">2</property>
<property name="Priority">1</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:3333</property>
<property name="SetupTime">2011-05-16_08:30:45.113+0000</property>
</properties>
```

When C answers call from A, A's call with B is disconnected.

**SET\_PARAMETER** (sent to indicate that call with B has been cleared)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:50.123+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>
```

**SET\_PARAMETER** (sent to indicate that call with C has been answered)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="ConnectedNr">tel:3333</property>
<property name="ConnectTime">2011-05-16_08:30:50.233+0000</property>
</properties>
```

**TEARDOWN** (sent when call with C is finished)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:58.173+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>
</properties>
```

**A.1.6. RTSP message sequence for Call intrusion description**

In the case of a call intrusion, arrival or establishment of the 1st call **SHALL** cause the VCS client to establish an RTSP session to the RTSP server. A SETUP message without XML body followed by a SET\_PARAMETER message with CRD XML body **SHALL** be sent to the RTSP server to establish an RTSP session.

When the 1st call is accepted, the RECORD message with XML body **SHALL** be sent to the RTSP.

When the 2nd call (intruding call) arrives, the VCS **SHALL** send a SET\_PARAMETER with CRD XML body to the RTSP server. This informs the RTSP server of a call intrusion request to connect with the 1<sup>st</sup> call. When the 2nd call is accepted, the VCS **SHALL** send another SET\_PARAMETER with CRD XML body to inform the RTSP server of this call accept. A sum of the audio from all parties in the conference **SHALL** now be sent to the RTSP server.

When a call (intruding or first call) is cleared, the VCS **SHALL** send a SET\_PARAMETER message with CRD XML body to the RTSP server. This will inform the RTSP server that the call intrusion or 3-party conference has stopped. The audio from the party leaving the 3-party conference will therefore no longer be sent to the RTSP server.

When the last call is cleared the VCS **SHALL** send a PAUSE followed by TEARDOWN message (or TEARDOWN message directly) with CRD XML body to the RTSP server. This will inform the RTSP server that the last call has been cleared. The audio from the call will therefore no longer be sent to the RTSP server.

The same principle as described above **SHALL** also be used for conference calls.

**A.1.7. Call intrusion example**

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when the 1<sup>st</sup> incoming call indicated by SIP INVITE has been received by the VCS conference focus. The RTSP server responds to the RTSP SETUP request with a 200 OK response.

The following messages are then sent to the RTSP server.

**SET\_PARAMETER** (sent after 200 OK response received to RTSP SETUP)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="CallRef">XXXXXXX-11111111@192.168.1.10</property>
<property name="Direction">1</property>
<property name="Priority">3</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:30.123+0000</property>
</properties>
```

**RECORD** (sent when call answered by sending 200 OK response to SIP INVITE)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="ConnectedNr">tel:2222</property>
<property name="ConnectTime">2011-05-16_08:30:35.123+0000</property>
</properties>
```

**SET\_PARAMETER** (sent when SIP INVITE for the intruding call is received)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="CallRef">XXXXXXX-22222222@192.168.1.20</property>
<property name="Direction">1</property>
<property name="Priority">1</property>
<property name="CallingNr">tel:3333</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:40.113+0000</property>
</properties>
```

**SET\_PARAMETER** (sent when intruding call answered by sending 200 OK response to SIP INVITE- parties now in 3-party conference)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="ConnectedNr">tel:2222</property>
<property name="ConnectTime">2011-05-16_08:30:44.233+0000</property>
</properties>
```

**SET\_PARAMETER** (sent when intruding call is cleared on receiving SIP BYE and removed from 3-party conference)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:50.173+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>
```

**TEARDOWN** (sent when original call cleared on sending or receiving SIP BYE)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:55.523+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>
```

#### A.1.8. Conference call example

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when the 1<sup>st</sup> incoming call indicated by SIP INVITE has been received by the VCS conference focus. The RTSP server responds to the RTSP SETUP request with a 200 OK response.

The following messages are then sent to the RTSP server.

**SET\_PARAMETER** (sent after 200 OK response received to RTSP SETUP)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="CallRef">XXXXXXXX-11111111@192.168.1.10</property>
<property name="ConfRef">YYYYYYYY-ZZZZZZZ@192.168.1.10</property>
<property name="Direction">1</property>
<property name="Priority">3</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:30.123+0000</property>
</properties>
```

**RECORD** (sent when call answered by sending 200 OK response to SIP INVITE)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="ConnectedNr">tel:2222</property>
<property name="ConnectTime">2011-05-16_08:30:35.123+0000</property>
</properties>
```

**SET\_PARAMETER** (sent when SIP INVITE for 2<sup>nd</sup> call is received)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="CallRef">XXXXXXXX-22222222@192.168.1.20</property>
<property name="ConfRef">YYYYYYYY-ZZZZZZZ@192.168.1.20</property>
<property name="Direction">1</property>
<property name="Priority">1</property>
<property name="CallingNr">tel:3333</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:40.113+0000</property>
</properties>
```

**SET\_PARAMETER** (sent when 2<sup>nd</sup> call answered by sending 200 OK response to SIP INVITE and added to conference)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="ConnectedNr">tel:2222</property>
<property name="ConnectTime">2011-05-16_08:30:44.233+0000</property>
</properties>
```

**SET\_PARAMETER** (sent when 2<sup>nd</sup> call is cleared on receiving SIP BYE and removed from conference)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:50.173+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>
```

**TEARDOWN** (sent when 1<sup>st</sup> call cleared on sending or receiving SIP BYE)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:55.523+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>
```

#### A.1.9. Call Hold operation example

This operation **MAY** be used to notify the RTSP server (Recorder equipment) that a call has been put on-hold, the UTC time of when the call was placed on hold and the party that placed the call on hold (i.e. Calling or Called party). It will also be used to notify the RTSP server when a call has been removed from hold. When a call is put on hold the audio stream being sent to the RTSP server **MAY** contain nothing/silence/music.

An example of the hold operation is as follows:

```
<operation name="HOLD" time="2011-05-16_08:30:33.123+0000">1</operation>
```

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when an outgoing call is initiated by sending a SIP INVITE. The RTSP server will respond to the RTSP SETUP request with a 200 OK response.

The following messages are then sent to the RTSP server.

**SET\_PARAMETER** (sent after 200 OK response received to RTSP SETUP)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="CallRef">XXXXXXX-11111111@192.168.1.10</property>
<property name="Direction">2</property>
<property name="Priority">1</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="SetupTime">2011-05-16_08:30:30.123+0000</property>
</properties>
```

**RECORD** (sent when call answered on receiving 200 OK response to sent SIP INVITE)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="ConnectedNr">tel:2222</property>
```

```
<property name="ConnectTime">2011-05-16_08:30:36.123+0000</property>
</properties>
```

```
SET_PARAMETER (sent when call placed on hold by calling party)
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<operations>
<operation name="HOLD">1</property>
</properties>
```

```
SET_PARAMETER (sent when call resumed)
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<operations>
<operation name="HOLD">0</property>
</properties>
```

```
TEARDOWN (sent when call cleared on sending or receiving SIP BYE)
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="DisconnectTime">2011-05-16_08:30:50.123+0000</property>
<property name="DisconnectCause">16</property>
<property name="DisconnectSource">1</property>
```

## A.2. CRD messages for Radio Calls

### A.2.1. PTT activation at VCS endpoint example

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when a SIP session request to the radio is initiated by sending a SIP INVITE. The RTSP server will respond to the RTSP SETUP request with a 200 OK response.

The following RTSP messages **SHALL** be sent to the RTSP server (Recorder equipment) at the VCS endpoint when there is a Normal-PTT activation and PTT deactivation event towards a radio transmitter/transceiver.

**RECORD** (on PTT activation)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="FrequencyID">100.100</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="Direction">2</property>
</properties>
<operations>
<operation name="PTT" time="2011-05-16_08:30:30.123+0000">1</operation>
</operations>
</call-record-data>
```

**PAUSE** (on PTT deactivation)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<operations>
<operation name="PTT" time="2011-05-16_08:30:43.123+0000">0</operation>
</operations>
</call-record-data>
```

### A.2.2. PTT activation at Radio endpoint example

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when a SIP session request to the radio is initiated by sending a SIP INVITE. The RTSP server will respond to the RTSP SETUP request with a 200 OK response.

The following RTSP messages **SHALL** be sent to the RTSP server (Recorder equipment) at the Radio endpoint when there is an Emergency-PTT activation and PTT deactivation event towards a radio transmitter/transceiver.

**RECORD** (on PTT activation)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<properties>
<property name="FrequencyID">200.200</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="Direction">1</property>
</properties>
<operations>
<operation name="PTT" time="2011-05-16_08:30:30.123+0000">4</operation>
```

```
</operations>
</call-record-data>
```

PAUSE (on PTT deactivation)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
<operations>
<operation name="PTT" time="2011-05-16_08:30:43.123+0000">0</operation>
</operations>
</call-record-data>
```

### A.2.3. SQU activation at VCS endpoint example

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when a SIP session request to the radio is initiated by sending a SIP INVITE. The RTSP server will respond to the RTSP SETUP request with a 200 OK response.

The following RTSP messages **SHALL** be sent to the RTSP server (Recorder equipment) at the VCS endpoint when there is a SQU activation and SQU deactivation event received from a radio receiver/transceiver.

RECORD (on SQU activation)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="FrequencyID">100.100</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="Direction">2</property>
</properties>
<operations>
<operation name="SQU" time="2011-05-16_08:30:30.123+0000">1</operation>
</operations>
</call-record-data>
```

PAUSE (on SQU deactivation)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<operations>
<operation name="SQU" time="2011-05-16_08:30:43.123+0000">0</operation>
</operations>
</call-record-data>
```

### A.2.4. SQU activation at Radio endpoint example

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when a SIP session request to the radio is initiated by sending a SIP INVITE. The RTSP server will respond to the RTSP SETUP request with a 200 OK response.

The following RTSP messages **SHALL** be sent to the RTSP server (Recorder equipment) at the radio endpoint when there is a SQU activation and SQU deactivation event sent from the radio receiver/transceiver.

RECORD (on SQU activation)

```
<call-record-data connref="22222222-XXXXXXX@192.168.1.20">
```

```

<properties>
<property name="FrequencyID">200.200</property>
<property name="CallingNr">tel:1111</property>
<property name="CalledNr">tel:2222</property>
<property name="Direction">1</property>
</properties>
<operations>
<operation name="SQU" time="2011-05-16_08:30:30.123+0000">1</operation>
</operations>
</call-record-data>

```

PAUSE (on SQU deactivation)

```

<call-record-data connref="22222222-XXXXXXXX@192.168.1.20">
<operations>
<operation name="SQU" time="2011-05-16_08:30:43.123+0000">0</operation>
</operations>
</call-record-data>

```

#### A.2.5. Simultaneous SQU received at VCS endpoint on different frequencies

When the first SIP session is initiated to Radio f1 RTSP session(s) **SHALL** be created with the RTSP server by sending SETUP.

Further SIP sessions initiated to other radios (i.e. Radio f2) **SHALL NOT** create new RTSP sessions with the RTSP server.

When the last remaining SIP session is terminated to a radio, the RTSP session(s) with the RTSP server **SHALL** be closed using the TEARDOWN message.

When the first SQU ON is received, the RECORD message **SHALL** be sent to the RTSP server. An Audio stream **SHALL** also be sent via RTP to the RTSP server(s).

When the second SQU ON arrives, a SET\_PARAMETER with CRD XML body **SHALL** be sent to the RTSP server. This informs the RTSP server of a new SQU on the existing RTSP session(s). The audio from both SQU activities **SHALL** be summed and sent to the RTSP server.

When the first SQU OFF is received, a SET\_PARAMETER message with CRD XML body **SHALL** be sent to the RTSP server. The audio sent to the RTSP server **SHALL** then only contain audio from the remaining SQU activity.

When the second SQU-OFF is received, a PAUSE message with CRD XML body **SHALL** be sent to the RTSP server. This will inform the RTSP server that no Aircraft call exists. The Audio stream **SHALL NOT** be sent to the RTSP server after PAUSE.

**A.2.6. Receiver Voting operation**

In the case of multiple SQU signals being simultaneously received from radio receivers/transceivers operating on the same frequency (i.e. Receiver voting group), this operation **SHOULD** be used to notify the RTSP server (Recorder equipment) which Rx audio stream has been selected by the voting algorithm and the UTC time of when the Voting action took place. It will also be used to notify the RTSP server about which Rx stream(s) haven't been selected by the voting algorithm.

An example of the voting operation is as follows:

```
<operation name="VOTING" time="2011-05-16_08:30:30.123+0000">1</operation>
```

In the following example the RTSP session with the RTSP server is initiated by sending RTSP SETUP to the RTSP server when a SIP session request to the first radio is initiated by sending a SIP INVITE. The RTSP server will respond to the RTSP SETUP request with a 200 OK response. SIP session requests to successive radios **SHALL NOT** cause further RTSP sessions to be established.

When a SQU connref is selected by voting, SET\_PARAMETER **SHOULD** be sent with voting=1 on that SQU connref and voting=2 on all other SQU connrefs that are active at that time.

The following RTSP messages **SHALL** be sent to the RTSP server (Recorder equipment) at the VCS endpoint in the case of simultaneous SQU signals being received at the position from different Rx. Note that the SQU1\_ON and SQU1\_OFF pair have the same connref and that this is different from the connref assigned to SQU2\_ON and SQU2\_OFF pair.

**A.2.6.1. Example of Simultaneous SQU received at VCS from different radio receivers with voting operation enabled.**

**RECORD** (sent at first SQU activation indicating it as voted signal)

```
<call-record-data connref="11111111-XXXXXXX@192.168.1.10">
<properties>
<property name="FrequencyID">100.100</property>
</properties>
<operations>
<operation name="SQU" time="2011-05-16_08:30:30.123+0000">1</operation>
</operations>
</call-record-data>
```

**SET\_PARAMETER** (sent at second SQU activation now indicating it as voted signal)

```
<call-record-data connref="22222222-YYYYYYY@192.168.1.20">
<properties>
<property name="FrequencyID">200.200</property>
</properties>
<operations>
<operation name="SQU" time="2011-05-16_08:30:33.123+0000">1</operation>
<operation name="VOTING" time="2011-05-16_08:30:33.123+0000">1</operation>
</operations>
```

</call-record-data>

**SET\_PARAMETER** (sent at second SQU activation indicating first SQU activation is no longer voted signal)

<call-record-data connref="11111111-XXXXXXX@192.168.1.10">

<operations>

<operation name="VOTING" time="2011-05-16\_08:30:33.123+0000">2</operation>

</operations>

</call-record-data>

**SET\_PARAMETER** (sent at first SQU deactivation)

<call-record-data connref="11111111-XXXXXXX@192.168.1.10">

<properties>

<property name="FrequencyID">100.100</property>

</properties>

<operations>

<operation name="SQU" time="2011-05-16\_08:30:40.123+0000">0</operation>

</operations>

</call-record-data>

**PAUSE** (sent at second SQU deactivation)

<call-record-data connref="22222222-YYYYYYY@192.168.1.20">

<properties>

<property name="FrequencyID">200.200</property>

</properties>

<operations>

<operation name="SQU" time="2011-05-16\_08:30:43.123+0000">0</operation>

</operations>

</call-record-data>

**ANNEX B: REFERENCES**

- [1] IETF RFC 768: "User Datagram Protocol", August 1980.
- [2] IETF RFC 791: "Internet Protocol", September 1981.
- [3] IETF RFC 793: "Transmission Control Protocol", September 1981.
- [4] IETF RFC 2119: "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, March 1997.
- [5] IETF RFC 2326: "Real Time Streaming Protocol (RTSP)", April 1998.
- [6] IETF RFC 2460: "Internet Protocol, Version 6 (IPv6) Specification", December 1998.
- [7] IETF RFC 3261: "SIP: Session Initiation Protocol", June 2002.
- [8] IETF RFC 3264: "An Offer/Answer Model with the Session Description Protocol (SDP)", June 2002.
- [9] IETF RFC 3265: "Session Initiation Protocol (SIP)-Specific Event Notification", June 2002.
- [10] IETF RFC 3311: "The Session Initiation Protocol (SIP) UPDATE Method", September 2002.
- [11] IETF RFC 3325: "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", November 2002.
- [12] IETF RFC 3398: "Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping", December 2002.
- [13] IETF RFC 3428: "Session Initiation Protocol (SIP) Extension for Instant Messaging", December 2002.
- [14] IETF RFC 3515: "The Session Initiation Protocol (SIP) Refer Method", April 2003.
- [15] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications", July 2003.
- [16] IETF RFC 3665: Session Initiation Protocol (SIP) Basic Call Flow Examples, December 2003.
- [17] IETF RFC 3666: "Session Initiation Protocol (SIP). Public Switched Telephone Network (PSTN) Call Flows", December 2003.
- [18] IETF RFC 3840: "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", August 2004.
- [19] IETF RFC 3891: "The Session Initiation Protocol (SIP) 'Replaces' Header", September 2004.
- [20] IETF RFC 3911: "The Session Initiation Protocol (SIP) 'Join' Header", October 2004.
- [21] IETF RFC 4145: "TCP-Based Media Transport in the Session Description Protocol (SDP)", September 2005.
- [22] IETF RFC 4235: "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", November 2005.
- [23] IETF RFC 4346: "The Transport Layer Security (TLS) Protocol - Version 1.1", April 2006.
- [24] IETF RFC 4566: "SDP: Session Description Protocol", July 2006.
- [25] IETF RFC 4571: "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", July 2006.
- [26] IETF RFC 4579: "Session Initiation Protocol (SIP) Call Control – Conferencing for User Agents", August 2006.
- [27] IETF SIPING Working Group; draft-ietf-sipping-service-examples-13: "Session Initiation Protocol Service Examples", July 2007.
- [28] ITU-T. Rec. G.711. "Pulse Code Modulation (PCM) of Voice Frequencies", November 1988.
- [29] ITU-T. Rec. G.728. "Coding of Speech at 16 Kbit/s Using Low-delay Code Excited Linear Prediction", September 1992.
- [30] ICAO Convention on International Civil Aviation, Annex 10, Volume II, Chapter 3.5: "General Procedures for Recording of ATC communications"
- [31] ICAO Convention on International Civil Aviation, Annex 10, Volume III, Part II, Chapter 4: "Aeronautical Speech Circuits"
- [32] ICAO Convention on International Civil Aviation, Annex 11, Chapter 1: "Definitions"
- [33] ICAO Convention on International Civil Aviation, Annex 11, Chapter 6: "Air Traffic Services"

- Requirements for Communications”
- [34] ICAO Convention on International Civil Aviation, Annex 5: “Units of Measurement to be Used in Air and Ground Operations”

**ANNEX C: ACRONYMS**

Ack	Acknowledge
AGVN	Air Traffic Services Ground Voice communications Network
A/G	Air/Ground
ANSP	Air Navigation Service Provider
AOR	Address of Record
ATA	Analogue Telephone Adapter
ATC	Air Traffic Control
ATIS	Air Terminal Information Service
ATM	Air Traffic Management
ATS	Air Traffic Services
ATSU	Air Traffic Service Unit
AVP	Audio/Video Profile
BSS	Best Signal Selection
callref	Call Reference
connref	Connection Reference
CRD	Call Record Data
CWP	Controller Working Position
DA	Direct Access
FAA	Federal Aviation Administration
G/G	Ground/Ground
GRS	Ground Radio Station
IA	Instantaneous Access
ICAO	International Civil Aviation Organization
IDA	InDirect Access
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU-T	International Telecommunication Union – Telecommunication standardization sector
LAN	Local Area Network
MSC	Message Sequence Chart
NTP	Network Time Server
OVR	Override Call
PCM	Pulse Code Modulation
PTT	Press-To-Talk
QoS	Quality of Service
Rec.	Recommendation
RFC	Request For Comments
RRC	Radio Remote Control (Radio Gateway)
RRCE	Radio Remote Control Equipment
RTCP	Real-time Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-time Streaming Protocol
Rx	Receive
SDP	Session Description Protocol
SessionID	Session Identifier
SIP	Session Initiation Protocol

SQU	Squelch
TCP	Transmission Control Protocol
ThreadRef	Thread Reference
TLS	Transport Layer Secure protocol
Tx	Transmit
UA	User Agent
UDP	User Datagram Protocol
URI	Universal Resource Identifier
UTC	Universal Time Co-ordinated
VCS	Voice Communications System
VHF	Very High Frequency
VoIP	Voice over the Internet Protocol
WAN	Wide Area Network
XML	Extensible Markup Language